User's Manual
V1.00

Micriμm
Press

Weston, FL 33326

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326
USA

www.micrium.com

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where Micriμm Press is aware of a trademark claim, the product name appears in initial capital letters, in all capital letters, or in accordance with the vendor's capitalization preference. Readers should contact the appropriate companies for more complete information on trademarks and trademark registrations. All trademarks and registered trademarks in this manual are the property of their respective holders.

Copyright © 2012 by Micriμm except where noted otherwise. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

μC/Probe and the accompanying files are sold "as is". Micriμm makes and customer receives from Micriμm no express or implied warranties of any kind with respect to the software product, documentation, maintenance services, third party software, or other services. Micriμm specifically disclaims and excludes any and all implied warranties of merchantability, fitness for a particular purpose, and non-infringement. Due to the variety of user expertise, hardware and software environments into which μC/Probe may be subjected, the user assumes all risk of using μC/Probe. The maximum liability of Micriμm will be limited exclusively to the purchase price.

600-uC-Probe-002

Micriμm Press

Table of Contents

Chapter 1 Introduction 5

Chapter 2 µC/Probe System Overview 7
2-1 µC/Probe Data Client 10

Chapter 3 µC/Probe Symbol Browser 13
3-1 Loading an ELF file 13
3-2 Browsing the ELF file 14

Chapter 4 µC/Probe Settings 17
4-1 General Settings 18
4-2 Communication Settings 19
4-2-1 Segger J-Link 20
4-2-2 TCP/IP 21
4-2-3 RS-232 22

Chapter 5 µC/Probe Workspace Explorer 23

Chapter 6 µC/Probe Toolbox 25
6-1 Linear Gauges 26
6-2 Horizontal Linear Gauges 27
6-3 Quadrant Gauges 27
6-4 Semicircle Gauges 28
6-5 Circular Gauges 28
6-6 Half Donuts 29
6-7 Cylinders 29
6-8 Charts 30
6-9 Numerics 30

6-10 Miscellaneous 31
6-11 Writable Controls 31

Chapter 7 µC/Probe Layout Design Tools 33
7-1 µC/Probe Example 35

Chapter 8 Associating Symbols to Virtual Controls and Indicators 37

Chapter 9 Run-Time Mode 41
9-1 Run-Time Checklist 41
9-2 Running µC/Probe and your Debugging Software at the same time 43

Appendix A Configuring Virtual Controls and Indicators 45
A-1 Virtual Indicators 46
A-2 Virtual Controls 50
A-3 Charts 57

Appendix B Kernel Awareness Screen 63

Appendix C License and Activation 67
9-3 Ordering 67
9-4 Activating 68

Appendix D Bibliography 69

Index 71

Chapter

1

Introduction

µC/Probe is a Windows application designed to read and write the memory of any embedded target processor during run-time. Memory locations are mapped to a set of virtual controls and indicators placed on a dashboard. Figure 1-1 shows an overview of the system and data flow.

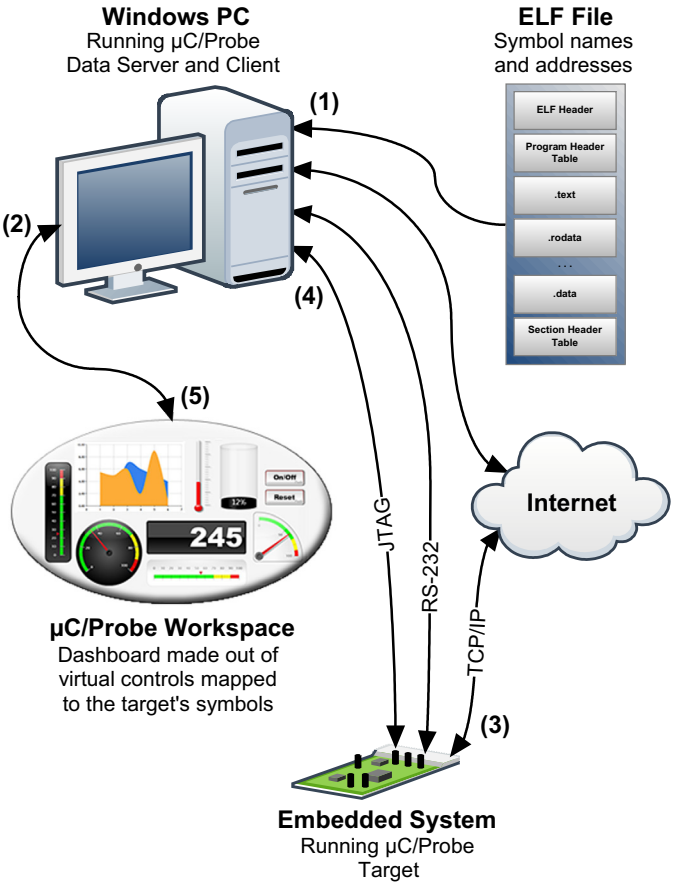

Figure 1-1 **µC/Probe Data Flow Diagram**

F1-1(1) You have to provide µC/Probe with an ELF file with DWARF-2 debugging information. The ELF file is generated by your toolchain's linker. µC/Probe parses the ELF file and reads the addresses of each of the embedded target's symbols (i.e. global variables) and creates a catalog known as *symbol browser*, which will be used by you during design-time to select the symbols you want to display on your dashboard. Refer to the document µC/Probe Target Manual for more information on installing the µC/Probe Target C files and building the ELF file.

F1-1(2) During design-time, you create a µC/Probe workspace using a Windows PC and µC/Probe. You design your own dashboard by dragging and dropping virtual controls and indicators onto a *data screen*. Each virtual control and indicator needs to be mapped to an embedded target's symbol by selecting it from the symbol browser. This document aims at providing more information on creating your own dashboard with µC/Probe.

F1-1(3) Before proceeding to the run-time stage, µC/Probe needs to be configured to use one of the three communication interfaces: JTAG, RS232 or TCP/IP. In order to start the run-time stage, you click the *Play* button and µC/Probe starts making requests to read the value of all the memory locations associated with each virtual control and indicator (i.e. buttons and gauges respectively). At the same time, µC/Probe sends commands to write the memory locations associated with each virtual control (i.e. buttons on a click event).

F1-1(4) In the case of a reading request, the embedded target responds with the latest value. In the case of a write command, the embedded target responds with an acknowledgement. Refer to the document µC/Probe Target Manual for more information on all you need in regards to the firmware that implements the communication interface that runs on the embedded target.

F1-1(5) µC/Probe parses the responses from the embedded target and updates the virtual controls and indicators.

Refer to the document µC/Probe Target Manual for more information about the firmware that resides on the Embedded System.

This document only provides information about the Windows PC side of the system.

Chapter

2

µC/Probe System Overview

This section provides an overview of the µC/Probe Windows Application.

Whenever you start µC/Probe in your Windows PC, three different modules are started: µC/Probe Automatic Updates and Licensing System, µC/Probe Data Client and µC/Probe Data Server as illustrated in Figure 2-1:

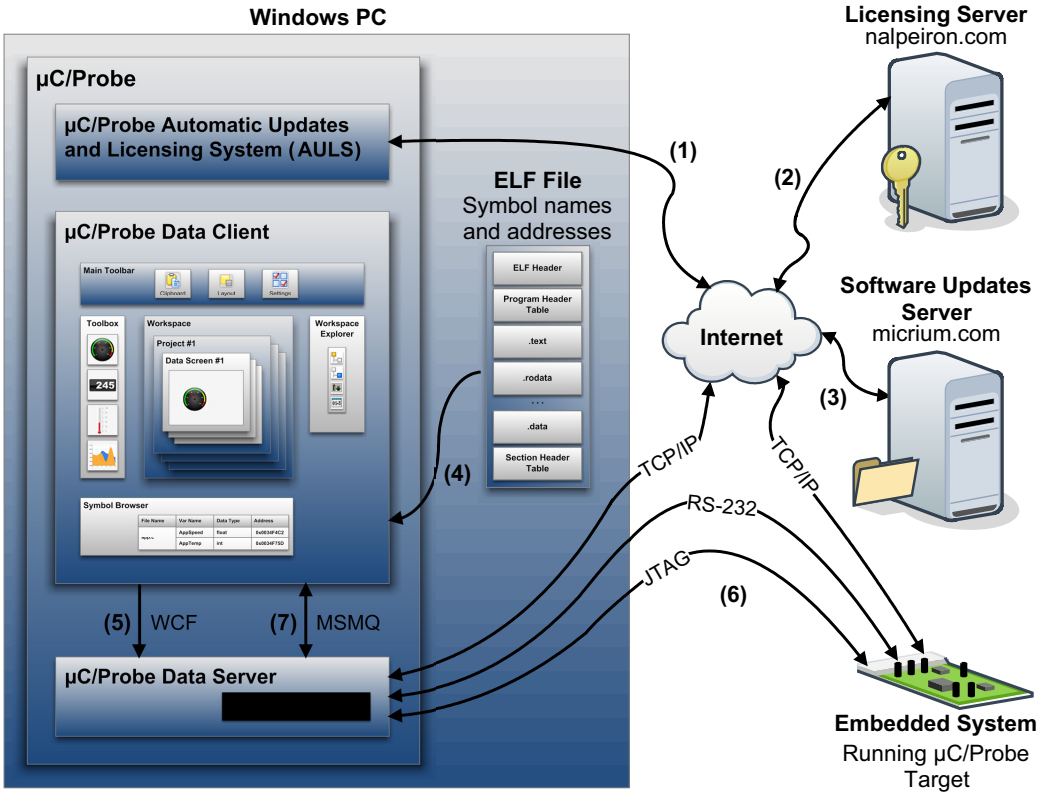

Figure 2-1 **µC/Probe System Overview**

F2-1(1) The Automatic Updates and Licensing System (AULS) is the part of the application that allows you to install and keep your µC/Probe application up to date.

F2-1(2) The trial version of µC/Probe is deployed with all the full version's features with the exception that your dashboard can only contain 10 symbols. Trial versions do not require internet access to activate the software application.

The full version of µC/Probe requires internet access to validate the license key provided by your Micriμm's sales representative. µC/Probe's licensing infrastructure is based on the Nalpeiron Licensing Service.

For more information on µC/Probe Licensing see Appendix C, "License and Activation" on page 67.

F2-1(3) Both versions of µC/Probe are self-updating and every time you start the application, if internet access is available, the µC/Probe AULS module checks for newer versions of µC/Probe from the Micriμm website and as they become available, the µC/Probe AULS module, automatically replaces any updated files.

F2-1(4) µC/Probe is based on a client/server model design where both, client and server reside in the same PC.

The µC/Probe Data Client is the part of the application that allows you to design your dashboard (design-time mode) and run it (run-time mode).

The next section in this document provides more information in regards to using the µC/Probe Data Client during design-time and run-time.

F2-1(5) The µC/Probe Data Server is the part of the application that runs in the background. You do not get to see it but it is the part of the application that connects directly with the Embedded Target and responds to the requests from the Data Client.

When the µC/Probe run-time mode gets started, the Data Client sends a message to the Data Server through a Windows Communication Foundation (WCF) channel. The message contains not only the embedded target communication settings but also all the symbol's memory address required by your dashboard design.

F2-1(6) The µC/Probe Data Server takes the message from the µC/Probe Data Client and initiates a communication with the embedded target through the configured communication interface.

F2-1(7) The µC/Probe Data Client establishes a connection with the µC/Probe Data Server through a Microsoft Message Queuing (MSMQ) port and exchanges requests to read and write the memory locations required by the current view of your dashboard's design.

2-1 µC/PROBE DATA CLIENT

The µC/Probe Data Client is illustrated in more detail in Figure 2-2:

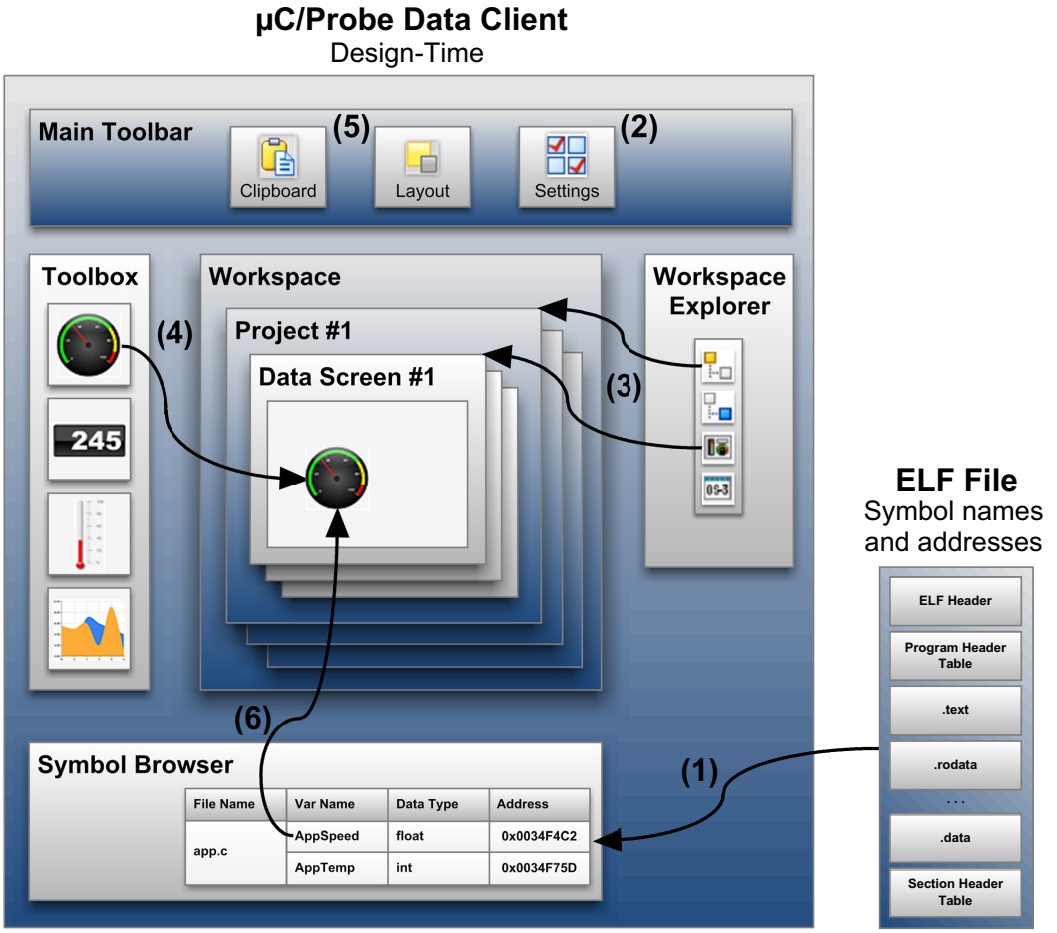

Figure 2-2 **µC/Probe Data Client: Design Time**

F2-2(1) The µC/Probe Data Client is the part of the application that during design-time takes the ELF file with DWARF-2 debugging information. As previously discussed, the ELF file is generated by your toolchain's linker. The µC/Probe Data Client parses the ELF file and reads the addresses of each of the embedded target's symbols (i.e. global variables) and creates a catalog known as symbol browser, which will be used by you during design-time to select the

symbols you want to display on your dashboard. Refer to the document µC/Probe Target Manual for more information on installing the µC/Probe Target C files and building the ELF file.

For more information, see Chapter 3, "µC/Probe Symbol Browser" on page 13.

F2-2(2) During design-time it is necessary to adjust the communication and other general settings. See Chapter 4, "µC/Probe Settings" on page 17 for more information on configuring µC/Probe.

F2-2(3) The Workspace Explorer in the µC/Probe Data Client allows you to add or delete Projects and Data Screens.

For more information, see Chapter 5, "µC/Probe Workspace Explorer" on page 23.

F2-2(4) The µC/Probe Toolbox displays icons for the virtual controls and indicators that you can add to your Data Screens. Each toolbox icon can be dragged and dropped onto the Data Screen to build your own dashboard.

For more information, see Chapter 6, "µC/Probe Toolbox" on page 25.

F2-2(5) The µC/Probe Layout Design Tools help you arrange the virtual controls and indicators on your data screen by speeding up the creation of your dashboard and making it look great.

For more information, see Chapter 7, "µC/Probe Layout Design Tools" on page 33.

F2-2(6) The last step during design-time is to map each virtual control and indicator in your Data Screen with an Embedded Target's memory location. The symbol browser allows you to quickly find the variable you want to display and then all you have to do is drag the variable from the symbol browser and drop it onto the virtual control or indicator of your choice.

See Chapter 8, "Associating Symbols to Virtual Controls and Indicators" on page 37 for more information on using the symbol browser to map virtual controls and indicators to the embedded target's memory locations.

The actual µC/Probe windows application is shown in Figure 2-3:

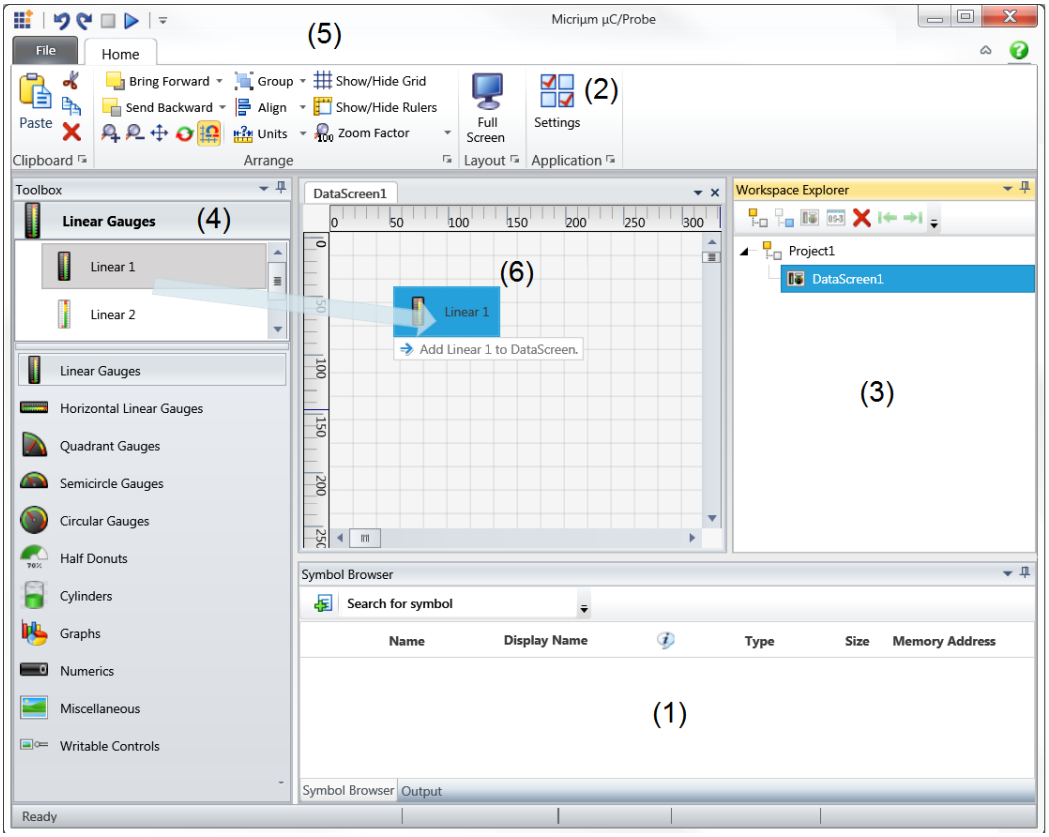

Figure 2-3 **µC/Probe Windows Application**

F2-3(1) Chapter 3, “µC/Probe Symbol Browser” on page 13.

F2-3(2) Chapter 4, “µC/Probe Settings” on page 17.

F2-3(3) Chapter 5, “µC/Probe Workspace Explorer” on page 23.

F2-3(4) Chapter 6, “µC/Probe Toolbox” on page 25.

F2-3(5) Chapter 7, “µC/Probe Layout Design Tools” on page 33.

F2-3(6) Chapter 8, “Associating Symbols to Virtual Controls and Indicators” on page 37.

Chapter

3

µC/Probe Symbol Browser

The µC/Probe's Symbol Browser is a list of your embedded target's symbols that helps you quickly find the symbol you want to use in your data screen. The symbol browser is available during design-mode and it is located at the bottom of the application window.

3-1 LOADING AN ELF FILE

In order to provide µC/Probe with the path of the embedded target's output file (ELF file), locate and click the button indicated in Figure 3-1:

Figure 3-1 **µC/Probe Symbol Browser: Loading an ELF file**

µC/Probe will prompt for the ELF file path by using a standard open file dialog. Locate the output file in your PC. The path is usually configured from your toolchain's linker configuration.

The ELF file needs to have symbolic information for debugging purposes in the DWARF-2 format.

Refer to the document µC/Probe-Target Manual for more information on building and configuring your toolchain to generate an ELF file.

3-2 BROWSING THE ELF FILE

µC/Probe parses the ELF file and creates a catalog that you can browse to search for the symbol you need. The symbol browser is a five-column tree list of symbols grouped by the C file name where the variable was declared as shown in Figure 3-2:

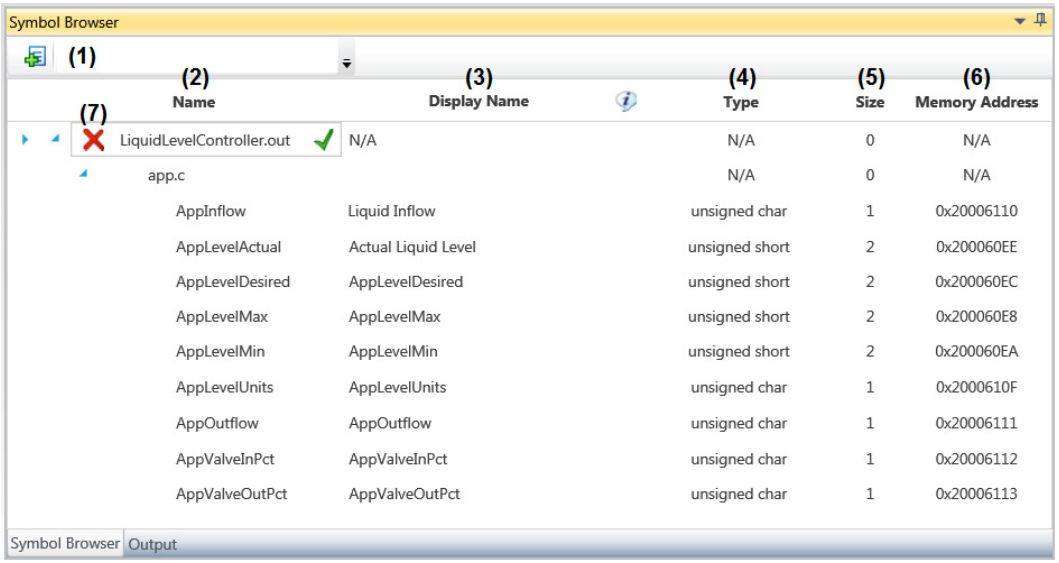

Figure 3-2 **µC/Probe Symbol Browser: Symbols grouped by C file**

F3-2(1) The symbol browser allows you to quickly find the symbol you want. Click on the symbol browser headers row to sort the list by the column you want. You can also expand and collapse tree nodes to focus on a particular C file, or you can use the search box.

F3-2(2) The **Name** column shows the name of the symbol as declared in your C file.

F3-2(3) The **Display Name** column by default displays the name of the symbol as declared in your C file, but also allows you to create an alias for the symbol. Double-click over the Display Name cell to create an alias.

F3-2(4) The **Type** column displays the symbol's C data type.

F3-2(5) The **Size** column displays the size in bytes of the symbol.

F3-2(6) The **Memory Address** column displays the symbol's location in the embedded target's memory.

F3-2(7) Click on the red **X** next to the name of the ELF file, to remove a symbol file from the symbol browser.

Chapter

4

µC/Probe Settings

The µC/Probe application's tool bar is located at the top of the application window.

The µC/Probe Settings window is opened by making click on the **Settings** button in the application's tool bar as indicated in Figure 4-1:

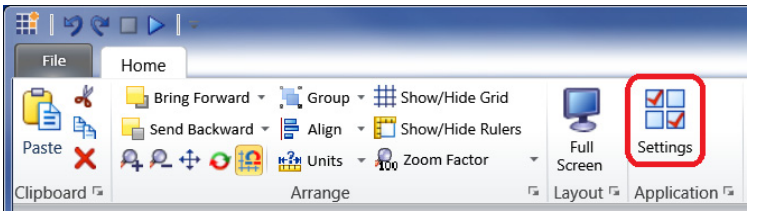

Figure 4-1 **µC/Probe Toolbar: Settings**

The settings window is divided in the following categories:

- General Settings:

  The general settings include the application debug logging, automatic updates and data server settings.

- Communication Settings:

  The communication settings includes the endianness type and the communication interface settings.

4-1 GENERAL SETTINGS

Figure 4-2 shows the µC/Probe General Settings window:

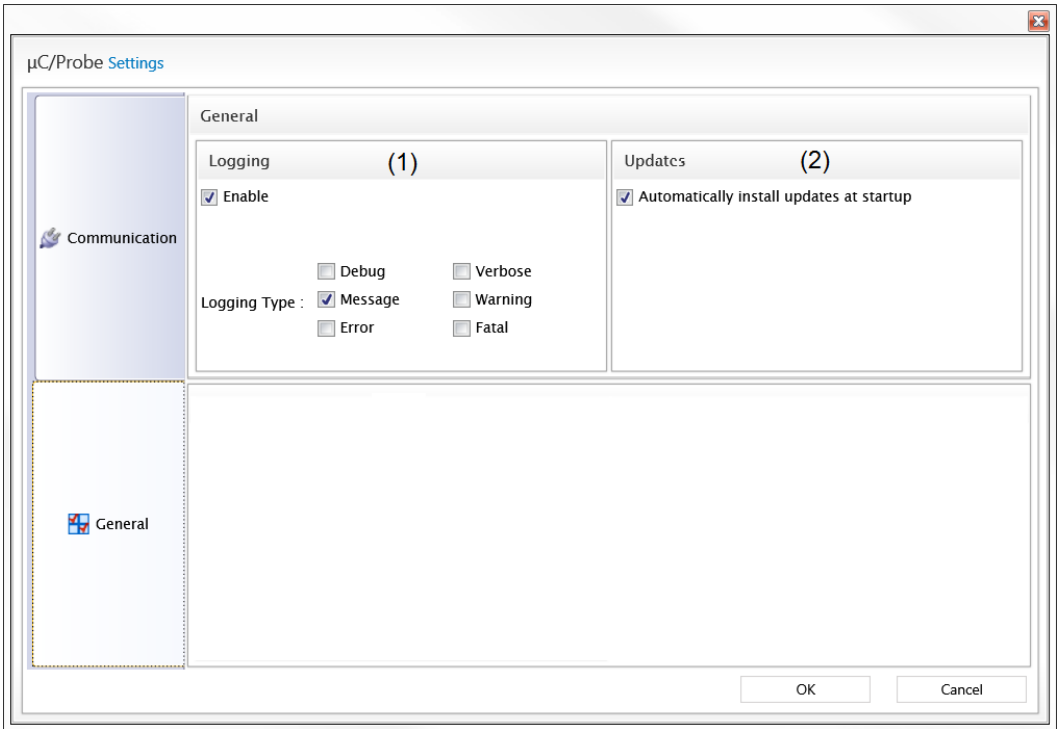

Figure 4-2 **µC/Probe General Settings**

F4-2(1) µC/Probe can be configured to log a specific level of verbosity for technical support purposes. If you ever have to contact Micriµm's technical support for any issues with your µC/Probe application, you can select a logging type that better describes your failure scenario.

F4-2(2) µC/Probe can be configured to automatically check and install updates as they become available from the Micriµm website. µC/Probe will check for software updates at startup if internet access is available.

4-2 COMMUNICATION SETTINGS

Figure 4-3 shows the µC/Probe communication settings window:

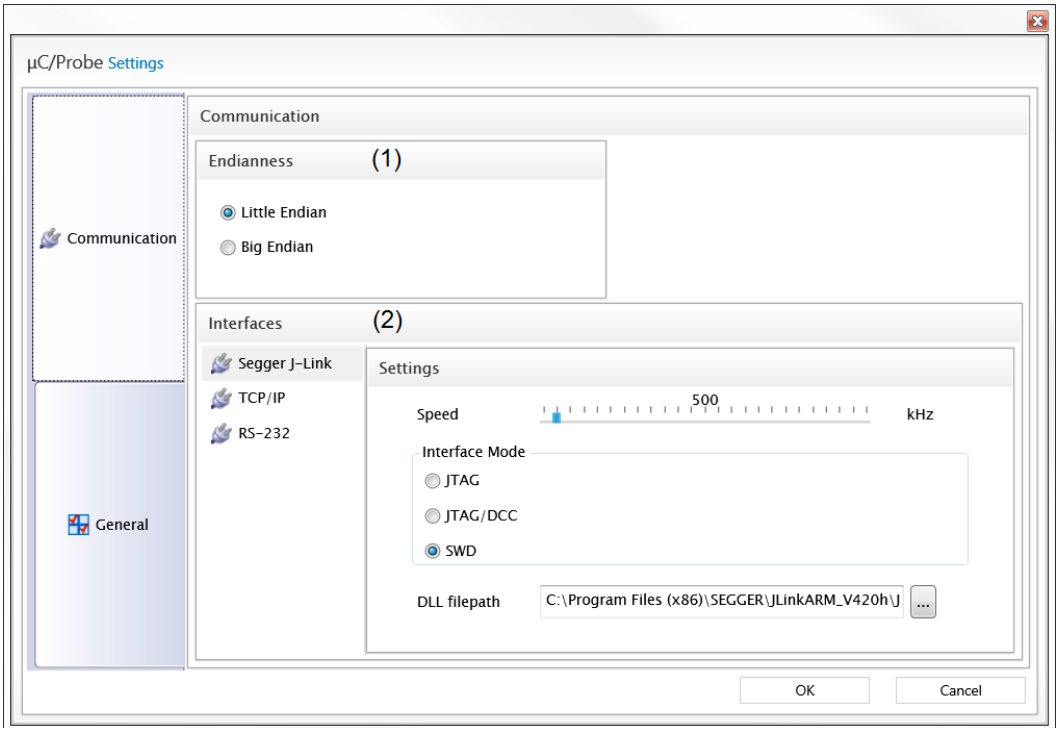

Figure 4-3 **µC/Probe Communication Settings**

F4-3(1) µC/Probe can be configured to interpret the byte order in either little endian or big endian, select the radio button that corresponds to your embedded target's endianness type.

F4-3(2) µC/Probe supports J-Link, TCP/IP and RS-232. Select the interface that your embedded target supports, and configure the settings corresponding to the interface.

The following sections describe how to configure each communication interface. Refer to the document µC/Probe-Target Manual for more information on the communication interface supported by the embedded target.

4-2-1 SEGGER J-LINK

J-Link is a USB powered JTAG emulator designed by Segger. In order to install the windows drivers for J-Link (J-Link DLL) go to Segger's website at www.segger.com and download the drivers from the appropriate download section.

J-Link is the most popular emulator for ARM cores and it does not require any special code resident in the embedded target to connect with µC/Probe.

If using J-Link, you can interface µC/Probe even with a bare-metal application running no kernel at all, as shown in Figure 4-4:

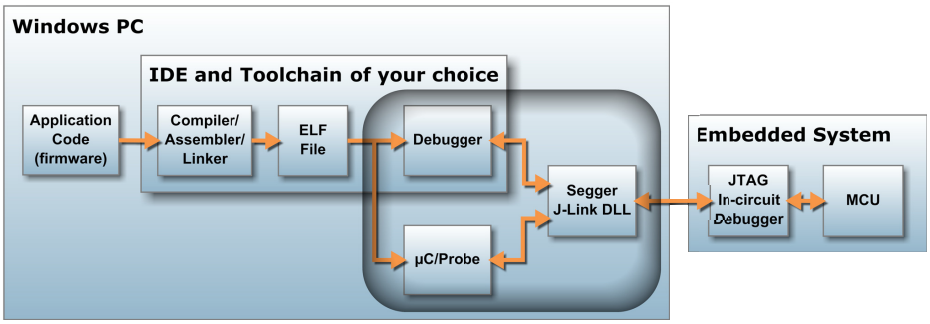

Figure 4-4 **µC/Probe via J-Link**

µC/Probe supports three J-link interface modes. Select the interface mode from the radio buttons and the path to the J-Link driver as shown in Figure 4-5:

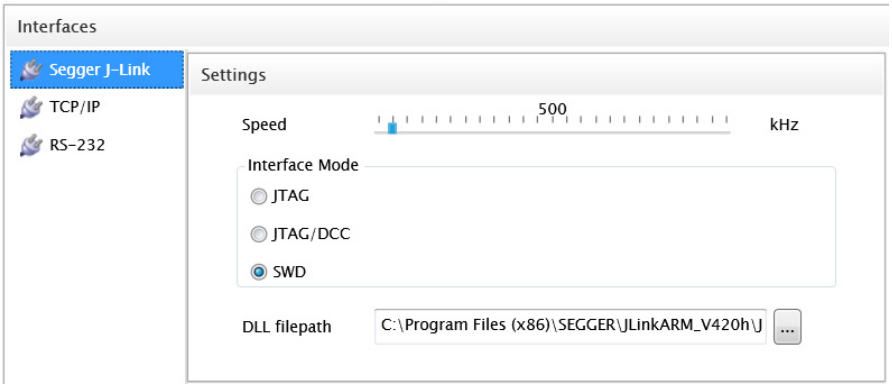

Figure 4-5 **µC/Probe Communication Settings: J-Link**

4-2-2 TCP/IP

µC/Probe supports a TCP/IP interface over the UDP protocol. This TCP/IP interface as opposed to J-Link, requires µC/Probe-Target code resident in your embedded system and because of the nature of TCP/IP, it also requires a kernel. Micriµm supports many cores and most likely this code is available from Micriµm. Contact Micriµm to find out if resident code for your particular setup is available.

Assuming your embedded system is running µC/Probe-Target as described in the document µC/Probe-Target Manual, enter the IP address and port number of your embedded system in the text boxes shown in Figure 4-6:

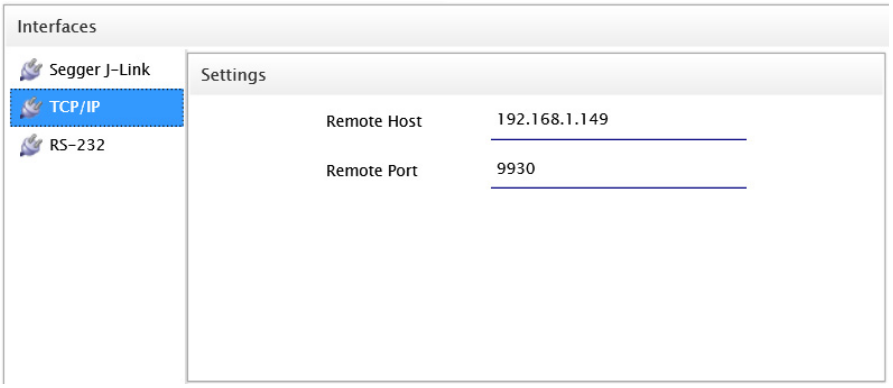

Figure 4-6 **µC/Probe Communication Settings: TCP/IP**

4-2-3 RS-232

µC/Probe supports a Serial RS-232 interface. This serial interface as opposed to J-Link, requires µC/Probe-Target code resident in your embedded system. Micriµm supports many UARTs and most likely this code is available from Micriµm. Contact Micriµm to find out if resident code for your particular setup is available.

Assuming your embedded system is running µC/Probe-Target as described in the document µC/Probe-Target Manual, enter the serial COM port number that your embedded target is attached to and select the baud rate from the drop downs shown in Figure 4-7:

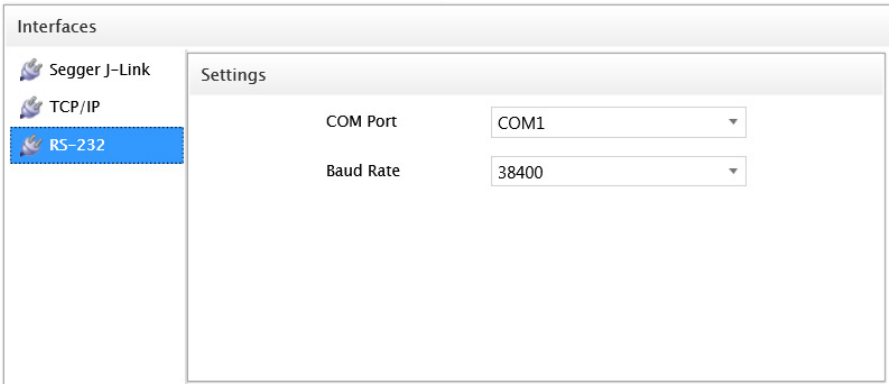

Figure 4-7 **µC/Probe Communication Settings: RS-232**

Chapter

5

µC/Probe Workspace Explorer

The µC/Probe Workspace Explorer is located on the right side of the application window and it is shown in Figure 5-1:

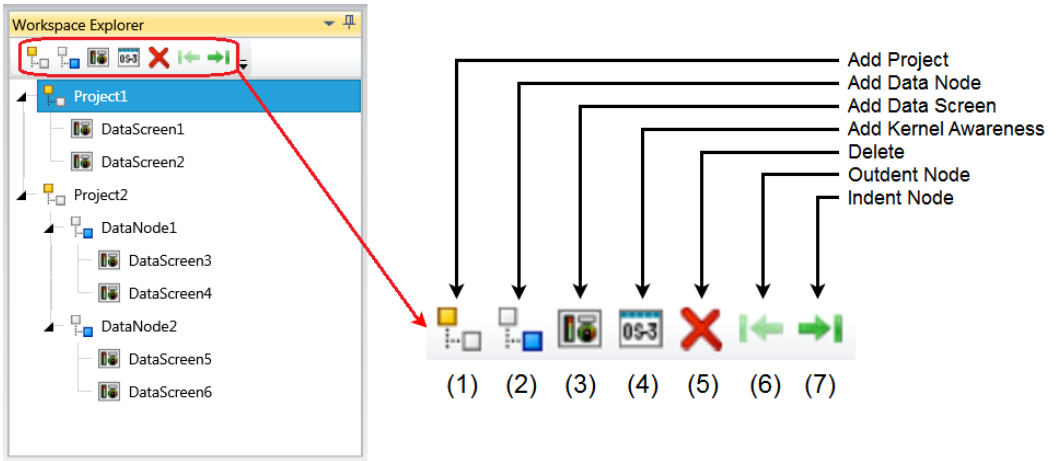

Figure 5-1 **µC/Probe Workspace Explorer**

F5-1(1) µC/Probe allows you to create a dashboard or user interface in a matter of minutes. The data screen is where you drag and drop the virtual controls and indicators. Your data screen's appearance and layout are very important. You can use Projects as a means to separate complex dashboards into multiple regions. Each project can contain multiple data screens and the workspace explorer allows you to navigate through this hierarchy.

F5-1(2) Similar to Projects, Data Nodes are just another level of hierarchy that allows you to group sets of virtual controls and indicators together into categories you define.

F5-1(3) Data Screens are the screens where you drag and drop the virtual controls and indicators. You can add as many data screens as you want.

F5-1(4) The Kernel Awareness Screen is a pre-configured Data Screen with all the symbols related to µC/OS-III. See Appendix B, “Kernel Awareness Screen” on page 63 for more information about this.

F5-1(5) Use the Delete button to delete an item from the workspace explorer, including Projects, Data Nodes, Data Screens and Kernel Awareness Screens.

F5-1(6) Use the indent button to push in an item in the Workspace Explorer tree. The items you can adjust the level of indentation include Projects, Data Nodes, Data Screens and Kernel Awareness Screens.

F5-1(7) Use the outdent button to push out an item in the Workspace Explorer tree. The items you can outdent include Projects, Data Nodes, Data Screens and Kernel Awareness Screens.

In order to organize your workspace tree you can also use your mouse to drag and drop items and rename items by invoking the context menu with a right-click.

Figure 5-2 shows an example of using projects and data nodes to better present a control panel for a liquid level control system:

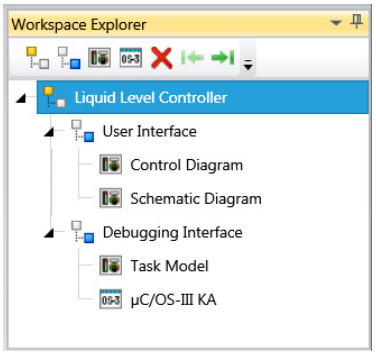

Figure 5-2 **Workspace Explorer for a Liquid Level Control System**

Chapter

6

µC/Probe Toolbox

The µC/Probe Toolbox is located on the left side of the application window and it is shown in Figure 6-1:

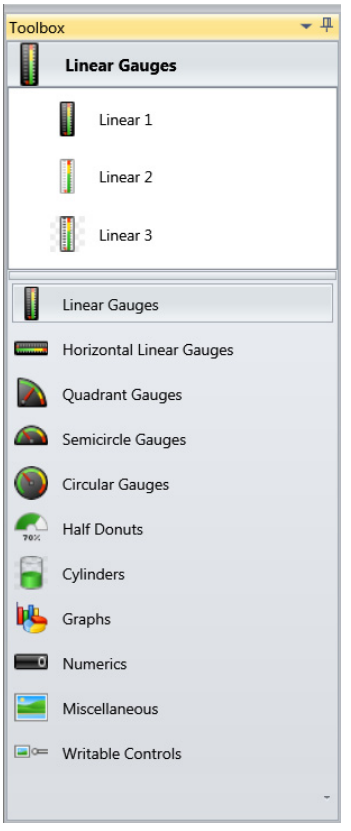

Figure 6-1 **µC/Probe Toolbox**

The items in the toolbox are contained in an accordion type of panel. You click on each button to display the items that belong to a category.

If you are running the full version of µC/Probe and have the automatic updates enabled, each category in this toolbox will expand with more virtual controls and indicators as software updates become available.

The following sections present a brief introduction to each of the toolbox categories. For more information on configuring each type of virtual control or indicator, see Appendix A, “Configuring Virtual Controls and Indicators” on page 45.

6-1 LINEAR GAUGES

Use the linear gauges shown in Figure 6-2 to display numeric data in a tri-color vertical scale. For more information configuring the properties of linear gauges see Appendix A, “Virtual Indicators” on page 46.

Figure 6-2 **µC/Probe Toolbox: Linear Gauges**

6-2 HORIZONTAL LINEAR GAUGES

Use the horizontal linear gauges shown in Figure 6-3 to display numeric data in a tri-color horizontal scale. For more information configuring the properties of horizontal linear gauges see Appendix A, “Virtual Indicators” on page 46.

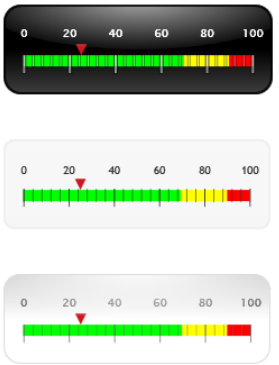

Figure 6-3 **µC/Probe Toolbox: Horizontal Linear Gauges**

6-3 QUADRANT GAUGES

Use the quadrant gauges shown in Figure 6-4 to display numeric data in a tri-color quadrant scale. For more information configuring the properties of quadrant gauges see Appendix A, “Virtual Indicators” on page 46.

Figure 6-4 **µC/Probe Toolbox: Quadrant Gauges**

6-4 SEMICIRCLE GAUGES

Use the semicircle gauges shown in Figure 6-5 to display numeric data in a tri-color semicircular scale. For more information configuring the properties of semicircle gauges see Appendix A, “Virtual Indicators” on page 46.

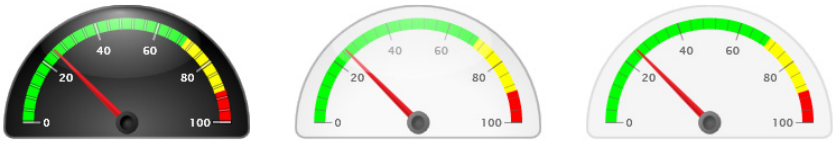

Figure 6-5 **µC/Probe Toolbox: Semicircle Gauges**

6-5 CIRCULAR GAUGES

Use the circular gauges shown in Figure 6-6 to display numeric data in a tri-color circular scale. For more information configuring the properties of circular gauges see Appendix A, “Virtual Indicators” on page 46.

Figure 6-6 **µC/Probe Toolbox: Circular Gauges**

6-6 HALF DONUTS

Use the half donut indicators shown in Figure 6-7 to display numeric data in a bi-color semicircular scale. For more information configuring the properties of half donuts see Appendix A, “Virtual Indicators” on page 46.

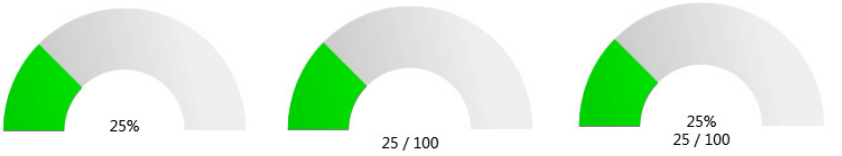

Figure 6-7 **µC/Probe Toolbox: Half Donuts**

6-7 CYLINDERS

Use the cylinder or tank level indicator shown in Figure 6-8 to display numeric data in a solid or gradient color. For more information configuring the properties of cylinders see Appendix A, “Virtual Indicators” on page 46.

Figure 6-8 **µC/Probe Toolbox: Cylinders**

6-8 CHARTS

Use the charts shown in Figure 6-9 to display numeric data including arrays in a marker, line or area chart. For more information configuring the properties of charts see Appendix A, “Charts” on page 57.

Figure 6-9 **µC/Probe Toolbox: Charts**

6-9 NUMERICS

Use the numeric indicators shown in Figure 6-10 to display numeric data in text. For more information configuring the properties of numeric indicators see Appendix A, “Formatting Properties Editor” on page 46 and Appendix A, “Numeric Indicator Properties Editor” on page 48.

Figure 6-10 **µC/Probe Toolbox: Numerics**

6-10 MISCELLANEOUS

Other miscellaneous indicators include a text box, thermometer and an image container capable of displaying an indexed array of images. For more information configuring the properties of these miscellaneous indicators see Appendix A, “Virtual Indicators” on page 46.

Figure 6-11 **µC/Probe Toolbox: Miscellaneous**

6-11 WRITABLE CONTROLS

The writable controls shown in Figure 6-12 include buttons, check boxes and sliders. Use these controls to modify the value of symbols from the embedded target. For more information configuring the properties of writable controls see Appendix A, “Virtual Controls” on page 50.

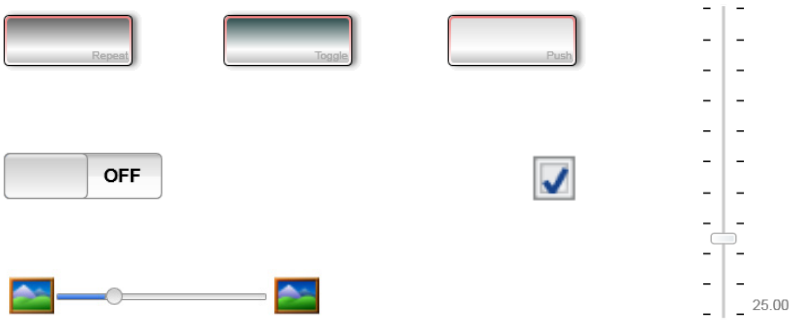

Figure 6-12 **µC/Probe Toolbox: Writable Controls**

Chapter

7

µC/Probe Layout Design Tools

The Layout Design Tools are located on the Main Toolbar at the top of the application's window. They include tools to arrange the virtual controls and indicators on your data screen as shown in Figure 7-1:

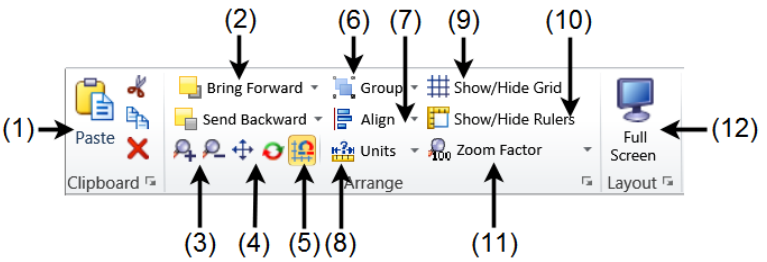

Figure 7-1 **µC/Probe Layout Design Tools**

F7-1(1) µC/Probe allows you to copy, cut and paste a single or multiple virtual controls or indicators. Everything you select will be stored temporarily into µC/Probe's clipboard during your session.

F7-1(2) µC/Probe allows you to drag and drop virtual controls and indicators onto the data screen and have them overlap one another. In some cases may be necessary to move some of them to the front of the data screen. Select the virtual control and indicator you want to move and click the **Bring Forward** or **Send Backward** button to move it to the front or to the back respectively.

F7-1(3) µC/Probe includes accessibility features that make the software more user friendly. You can control the zoom level of your µC/Probe Data Screen during both design-time and run-time. Click the buttons with the magnifier glass to zoom in and out, or use the keyboard shortcuts **Ctrl+** to zoom-in and **Ctrl-** to zoom-out.

F7-1(4) The panning tool in µC/Probe makes it easy to move a data screen around while your are zoomed in. The **Pan** button is a toggle button, clicking the button toggles the panning mode on and off.

F7-1(5) When you drag and drop a virtual control or indicator onto the data screen, the next thing you usually do is resize or move the object around the data screen. Turn the Snap-to-Grid mode on in order to align the virtual control or indicator to the nearest intersection of grid lines. The **Snap to Grid** button is a toggle button, clicking the button toggles the snap-to-grid mode on and off.

F7-1(6) You can combine multiple virtual controls and indicators so you can work with them as though they were a single object. You can resize, move, copy and paste all virtual controls and indicators in a group as a single unit.

After you have grouped virtual controls and indicators, you can still select any single object within the group without un grouping by first selecting the group, and then clicking on the object you want to select.

F7-1(7) µC/Probe allows you to easily align virtual controls and indicators by first selecting the group of objects you want to align and then clicking on one of the following alignment options:

- Left or Right Edges
- Top or Bottom Edges
- Horizontal or Vertical Centers

All the objects are aligned with respect to the first selected item.

F7-1(8) Use the **Units** button to select the grid and ruler's metric system.

F7-1(9) Use the **Show/Hide Grid** button to show and hide the grid lines on the data screen. The snap-to-grid mode still works even if the grid is not visible.

F7-1(10) Use the **Show/Hide Rulers** button to show or hide the ruler. The Show/Hide Rulers button is a toggle button, clicking the button turns the rulers on and off.

F7-1(11) Every time you click the magnifier glass buttons to zoom in and out, µC/Probe zooms in and out by certain zooming factor. Click the **Zoom Factor** button to select a different zooming factor.

F7-1(12) Click the **Full Screen** mode button to hide all the tools except the data screen. The Full Screen button is a toggle button, clicking the button turns the full screen mode on and off.

7-1 µC/PROBE EXAMPLE

In order to demonstrate the previous layout design tools, Figure 7-2 shows an example of a power plant's diagram used as a background to create a control panel with µC/Probe:

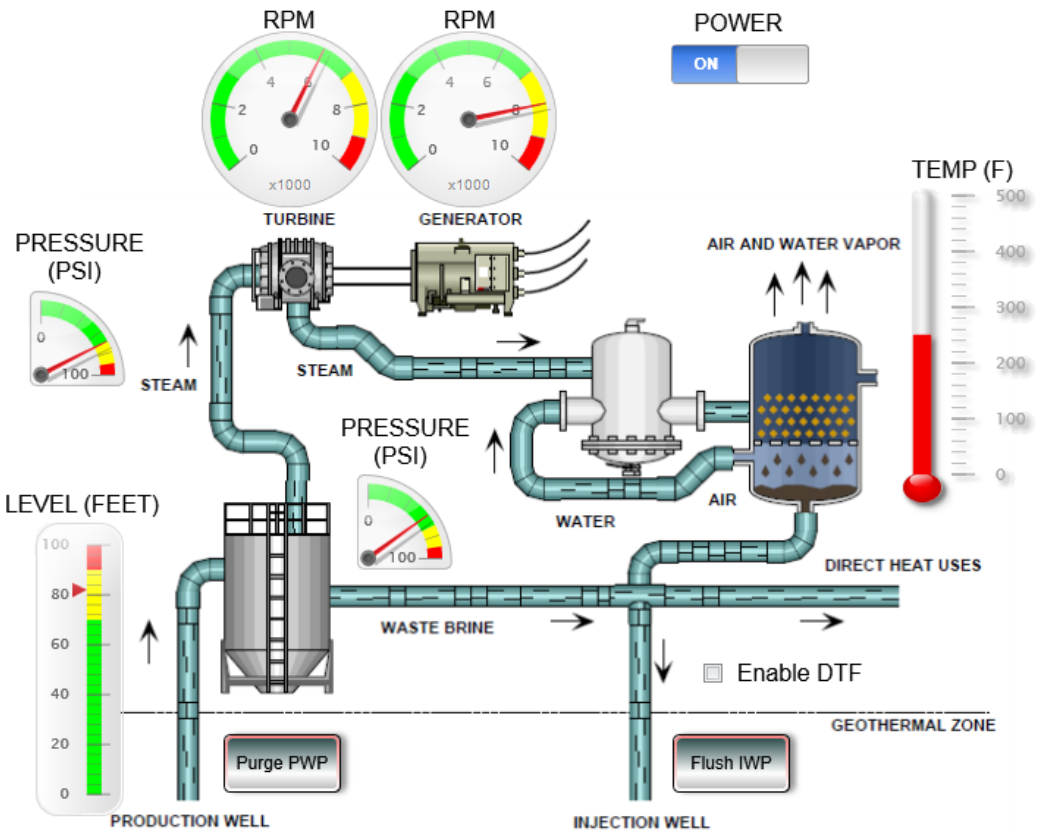

Figure 7-2 **µC/Probe Example of a Power Plant**

Chapter

8

Associating Symbols to Virtual Controls and Indicators

During design-time, use the µC/Probe symbol browser discussed in Chapter 3, “µC/Probe Symbol Browser” on page 13, to search and select the embedded target variables you want to associate to each of the virtual controls and indicators you placed on your data screen.

Once you find the symbol you want to associate, drag and drop the symbol over the virtual control or indicator you want, as shown in Figure 8-1:

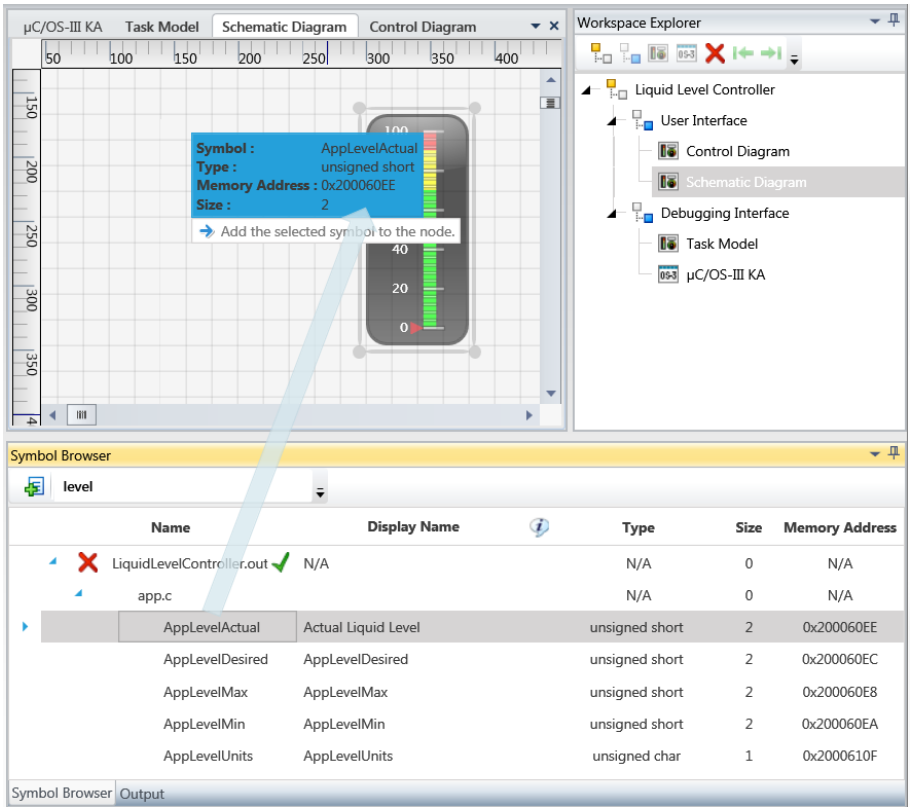

Figure 8-1 **Associating Symbols to Virtual Controls and Indicators**

Repeat the same process for each of the virtual controls and indicators placed on your data screen and µC/Probe will be ready to go into Run Mode unless you want to further configure other optional settings.

In order to configure other optional settings you can use the *symbols manager* by hovering your mouse pointer over the virtual control or indicator and making click on the icon shown in Figure 8-2:

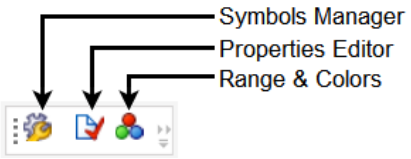

Figure 8-2 **Invoking the Symbols Manager**

Figure 8-3 shows the Symbols Manager:

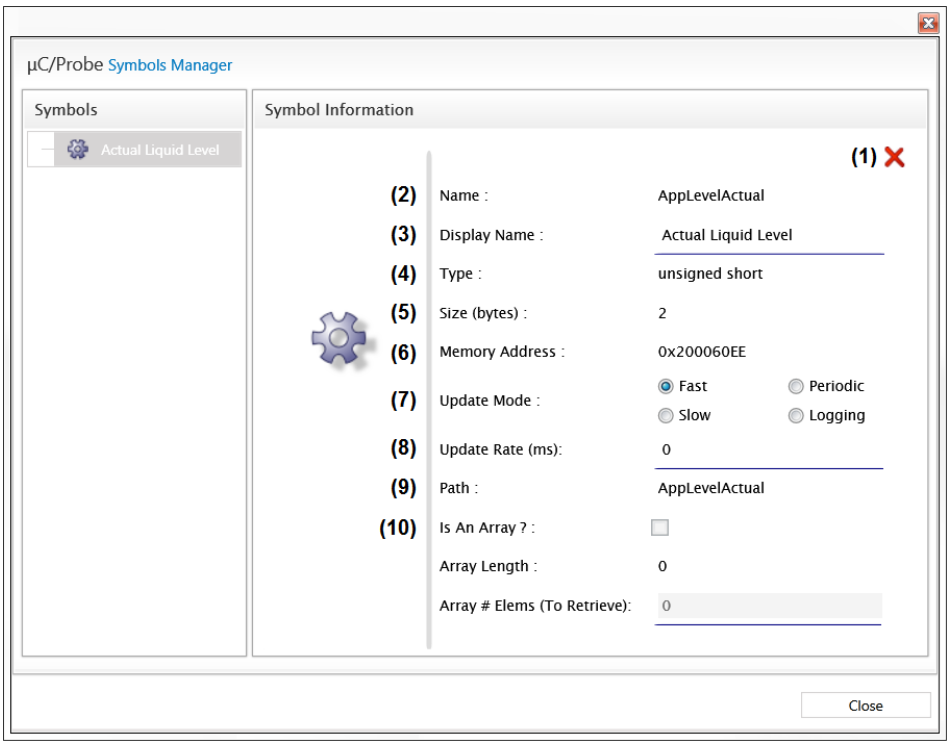
µC/Probe Symbols Manager
Symbols
Actual Liquid Level
Symbol Information
(1)
(2) Name : AppLevelActual
(3) Display Name : Actual Liquid Level
(4) Type : unsigned short
(5) Size (bytes) : 2
(6) Memory Address : 0x200060EE
(7) Update Mode : Fast Periodic Slow Logging
(8) Update Rate (ms): 0
(9) Path : AppLevelActual
(10) Is An Array ? :
Array Length : 0
Array # Elems (To Retrieve): 0
Close

Figure 8-3 **Symbols Manager**

F8-3(1) Click the red X to remove the symbol's association from the virtual control or indicator.

F8-3(2) Displays the name of the variable as declared in your C file.

F8-3(3) By default displays the name of the variable as declared in your C file, but this text box allows you to create an alias for display purposes.

F8-3(4) Displays the data type of the variable as declared in your C file.

F8-3(5) Displays the size of the variable in number of bytes.

F8-3(6) Displays the memory location of the variable in the embedded target's memory.

F8-3(7) You can select the update mode in a per object basis:

- Fast Mode: The µC/Probe Data Server will update this symbol as fast as it can.
- Slow Mode: The µC/Probe Data Server will update the symbol at 50% of the fastest speed possible. Use this update mode when trying to free up CPU resources.
- Periodic Mode: The µC/Probe Data Server will try to update the symbol at a specific update rate.

F8-3(8) The update rate applies to the Periodic Mode. Enter the desired update rate in milliseconds and the µC/Probe Data Server will try to keep the symbol up to date at this rate.

F8-3(9) The variable path displays the full variable name in those cases where the variable you selected is a member of a data structure.

F8-3(10) In case the variable is an array, you can enable indexing of just a certain amount of data.

Chapter

9

Run-Time Mode

9-1 RUN-TIME CHECKLIST

Before setting µC/Probe in Run-Time mode you should verify each of the following items:

Item #	Description	Reference
1	The embedded target has been programmed with an output file (ELF file) with debug information in the DWARF-2 format.	µC/Probe Target Manual: Chapter 5, on page 16
2	The embedded target is running and connected to the Windows PC through the communication interface of your choice.	µC/Probe Target Manual: Chapter 3, on page 10
3	µC/Probe has been configured with the latest output file (ELF file) that the embedded target is actually running.	µC/Probe User's Manual: Chapter 3, on page 13
4	µC/Probe has been configured with the proper communication interface and settings.	µC/Probe User's Manual: Chapter 4, on page 17
5	µC/Probe contains at least one virtual control or indicator on the data screen.	µC/Probe User's Manual: Chapter 7, on page 33
6	µC/Probe has been configured to associate the virtual control or indicator with one of the embedded target's variables displayed in the symbol browser.	µC/Probe User's Manual: Chapter 8, on page 37

Table 9-1 **Run-Time Mode Checklist**

In order to set µC/Probe in Run-Time mode, click on the **run** button indicated in Figure 9-1:

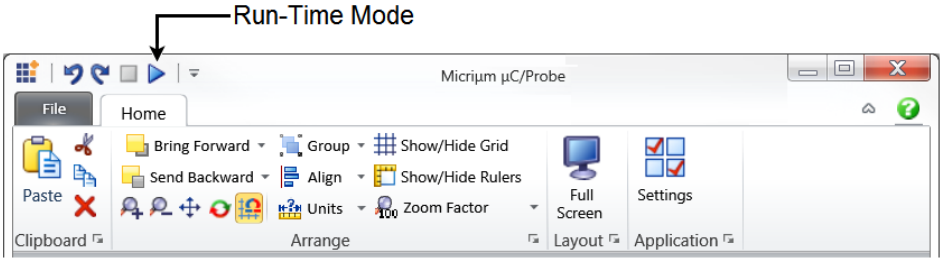

Figure 9-1 **Setting µC/Probe in Run-Time mode**

µC/Probe should start updating your data screens immediately and the application displays all kinds of status information in the status bar at the bottom of the µC/Probe window as shown in Figure 9-2:

Data server configured and processing requests
Segger J-Link (SWD) - 500 kHz
Connection to target established
Symbols Configured: 80 Rx: 6637 bytes/sec Tx: 6637 bytes/sec

Figure 9-2 **µC/Probe Status Bar**

You can also look at the µC/Probe output window during run-time as shown in Figure 9-3 to learn more about the system status:

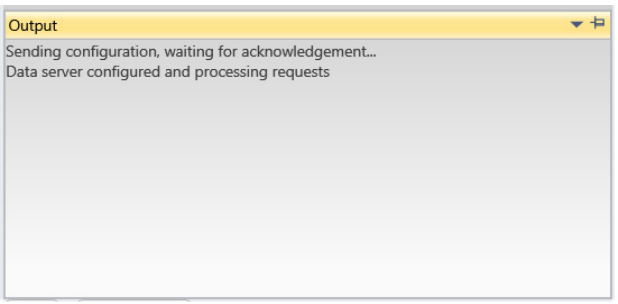

Figure 9-3 **µC/Probe Output Window**

9-2 RUNNING µC/PROBE AND YOUR DEBUGGING SOFTWARE AT THE SAME TIME

Your debugging software for embedded applications usually comes integrated with your IDE and at a minimum, allows you to step through the code, set breakpoints, display register and memory windows, display call stack information, and monitor variables and expressions. Examples of debugging software include IAR's C-SPY and GNU's GDB.

You can also use µC/Probe to extend the capabilities of your debugging software by running both at the same time. µC/Probe allows you to have instant control over your global variables in a real-time and non-intrusive way. From you debugger software, you can set breakpoints at locations of particular interest in the application being debugged and µC/Probe will stop updating the virtual controls and indicators at the same time.

This feature is accomplished by sharing the connection between the Windows PC and the Embedded Target being debugged. Whether the debugger of your choice is IAR's C-SPY, GNU's GDB or any other debugging software that supports J-Link, Figure 9-4 illustrates and example of running µC/Probe and the debugger of your choice at the same time:

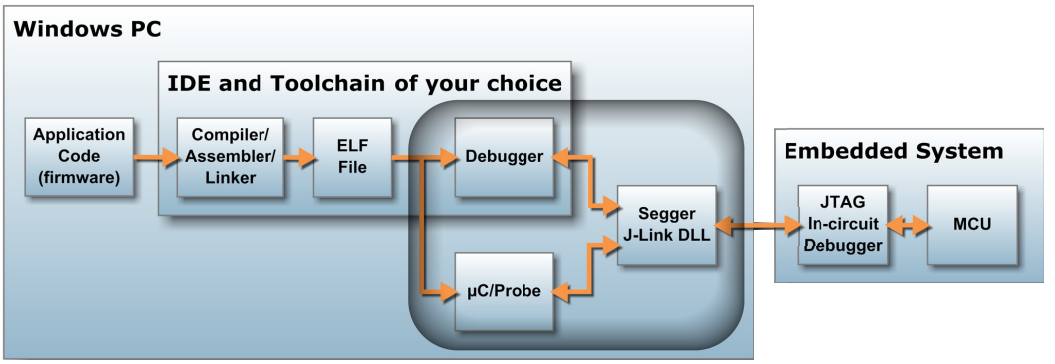

Figure 9-4 **Running µC/Probe and your debugging software at the same time**

Notice how µC/Probe and the Debugger Software not only share the same ELF file but also the same logic and physical interface through the Segger's J-Link DLL and JTAG in-circuit debugger respectively.

It is very important to make sure that both, µC/Probe and IAR's C-SPY are loading the same J-Link DLL as indicated in Figure 9-4.

In order to make sure that the same DLL is being loaded by both applications, whenever you set µC/Probe into Run-Time mode through a J-Link interface, µC/Probe will check for any active connection and will prompt you to select which of the active J-Link connections you want to use as shown in Figure 9-5:

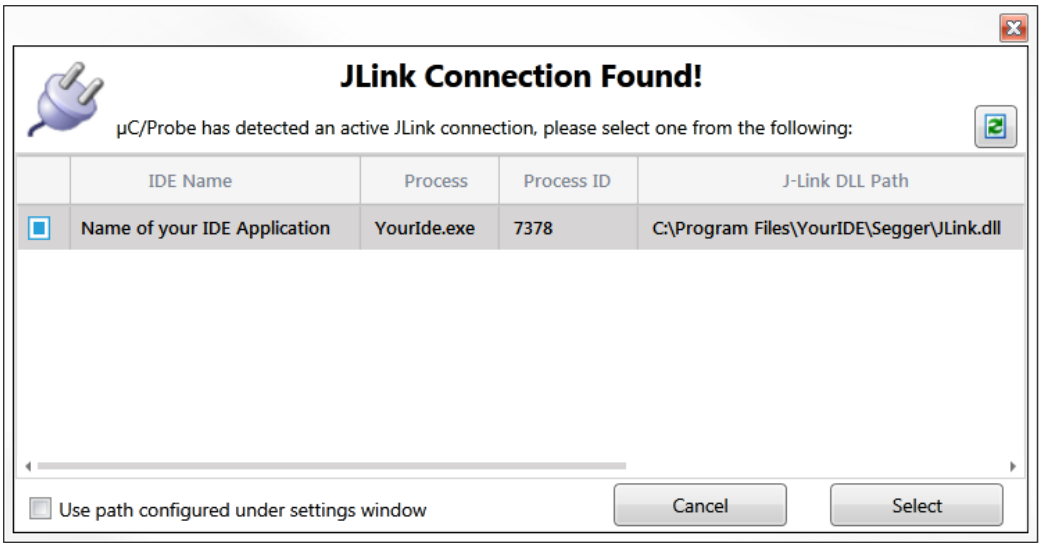

Figure 9-5 **Active J-Link connections**

By selecting one of the active connections you are specifying the path to the same J-Link DLL currently loaded by the IDE of your choice. The check box on the left corner at the bottom of the window allows you to override the DLL path with the one from the settings window as discussed in Chapter 4, “Segger J-Link” on page 20.

Appendix

A

Configuring Virtual Controls and Indicators

Once you drag and drop one of the virtual controls or indicators onto the data screen and associate it with one of the embedded target's symbols from the symbol browser, you can access the properties tool bar by moving the mouse over the virtual control or indicator.

The tool bar shown in Figure A-1 appears for you to select between one of the three configuration categories:

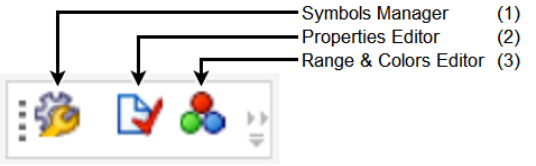

Figure A-1 **Virtual Controls and Indicators Toolbar**

FA-1(1) The Symbols Manager is common for all virtual controls and indicators, see Chapter 8, "Associating Symbols to Virtual Controls and Indicators" on page 37 for more information on the Symbols Manager.

FA-1(2) The Properties Editor is similar among most of the virtual controls and indicators and the next sections will describe how to use the Properties Editor for just a few of the most representative virtual controls and indicators.

FA-1(3) The Range and Colors Editor is only available to those virtual indicators that feature a multi-color scale. The next sections will describe how to use the Range and Colors Editor for a few of the most representative virtual indicators.

A-1 VIRTUAL INDICATORS

A-1-1 FORMATTING PROPERTIES EDITOR

The virtual indicators formatting category applies to linear gauges, half donuts, cylinders, numeric indicators, thermometers, graphs and any virtual indicator capable of showing the symbol's value in a graphical or text format. Figure A-2 shows the formatting category of a linear gauge:

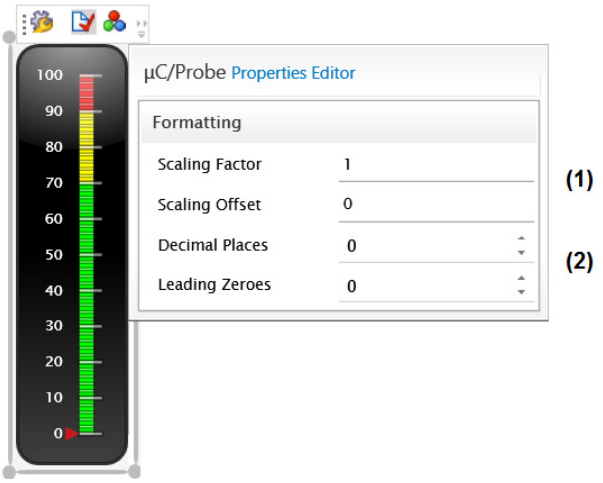

Figure A-2 **Formatting Properties Editor**

FA-2(1) In case you need to convert the value to Engineering Units (EU) before displaying in the virtual indicator, you can use the scaling factor and offset to specify the parameters of a linear conversion function. For example, if the embedded target's symbol you need to display is a 4-20mA value, you can implement the standard linear equation y = mx + b where m is the scaling factor, x is the 4-20mA value, b is the offset and y is the resulting Engineering Units (EU) value to display.

FA-2(2) During run-time mode, whenever you move the mouse pointer over a virtual indicator such as a linear gauge, a small yellow box will appear with the symbol's value formatted with the number of decimal places and leading zeros of your choice.

A-1-2 RANGE AND COLORS EDITOR

The Range and Colors Editor applies to linear gauges, half donuts, cylinders and any virtual indicator capable of displaying the symbol's value in a graphical format along a multi-color scale. Figure A-3 shows the Range and Colors Editor for a linear gauge:

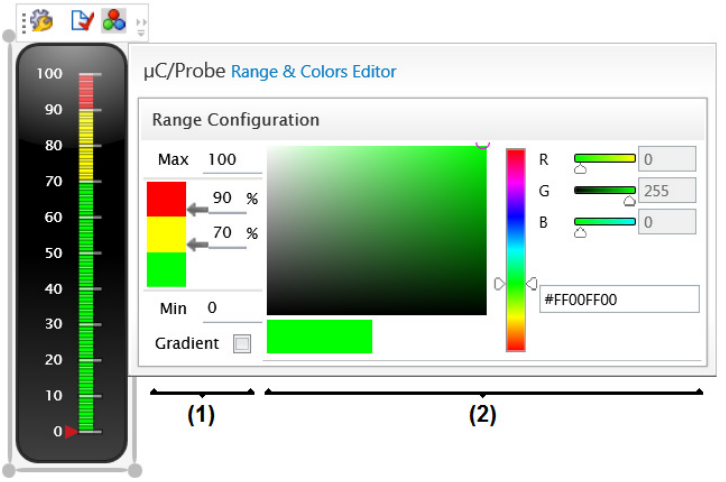

Figure A-3 **Range and Colors Editor**

FA-3(1) Start by setting the **Min** and **Max** limits, then click on the text boxes next to the percentage signs and enter the thresholds in terms of percentage.

FA-3(2) Each time you click on one of the text boxes to set the percentages, the color picker allows you to choose the color for that gauge band. You can enter the color you want in hex format or by selecting a color from the vertical slider and then fine tuning with the palette.

A-1-3 NUMERIC INDICATOR PROPERTIES EDITOR

The Numeric Indicator category from the Properties Editor only applies to numeric indicators. Figure A-4 shows the numeric indicator's properties. Font styles, alignment and the thousand separator, they all apply to the number 0 shown in white:

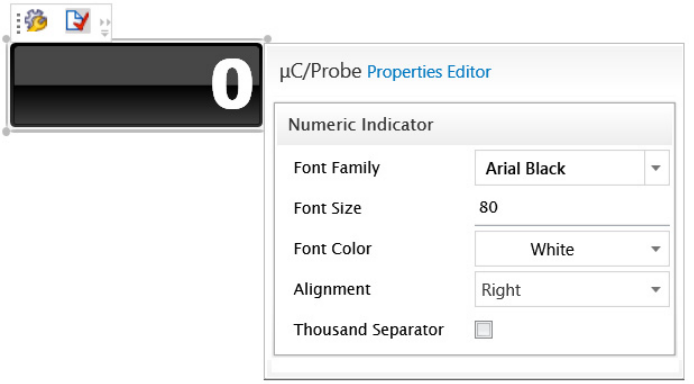

Figure A-4 **Numeric Indicator Properties Editor**

A-1-4 BITMAP ANIMATION PROPERTIES EDITOR

The bitmap animations are part of the toolbox's miscellaneous family. They are one of the most powerful virtual indicators because you have the freedom to customize it however you want by providing your own images.

Imagine you want to display the state of a valve to be either open or closed in a graphical way by using the bitmap images shown in Figure A-5 and an embedded target's application variable named `AppValveOutPct` that stores the state of the outflow valve (0%:open and 100%:closed).

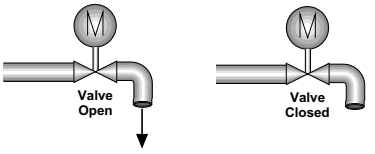

Figure A-5 **Bitmaps to Animate**

Figure A-6 shows the properties editor for the bitmap animation:

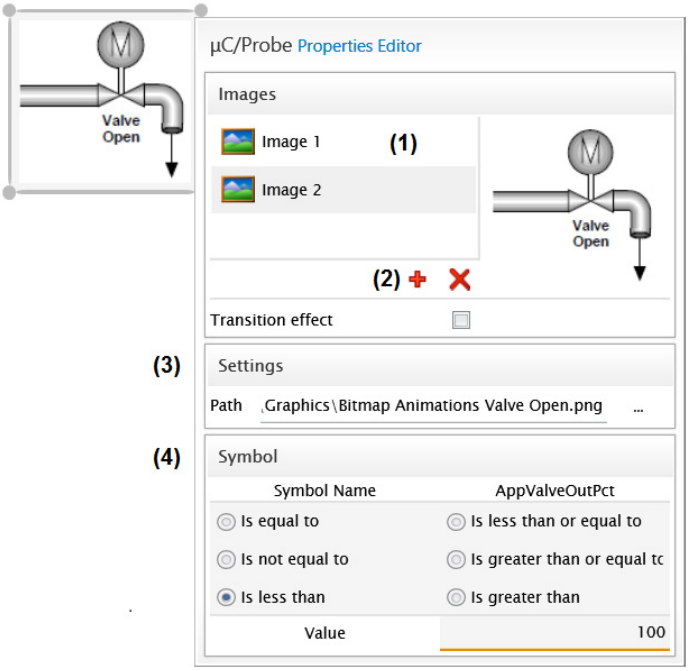

Figure A-6 **Bitmap Animation Properties Editor**

FA-6(1) The list of images is initialized with two images. Start by selecting the image you want to work with. A preview of the image is shown on the right side.

FA-6(2) You can add or delete more images into the list by making click on the red **+** or **x** buttons respectively.

Select the transition effect check box if you want to add a fade-in and fade-out effect between image transitions.

FA-6(3) Specify the bitmap file path or browse to it.

FA-6(4) Specify the rules. In this case, if `AppValveOutPct < 100` then the valve is open, and if `AppValveOutPct = 100` then the valve is all the way closed.

A-2 VIRTUAL CONTROLS

A-2-1 SLIDER CONTROL PROPERTIES EDITOR

The slider control is one of the few writable controls. It allows you to gradually modify an adjustable embedded target symbol's value. The user gets to select from a range of values by moving a value indicator up and down a track. For example, you typically create a volume control by using a slider control.

Figure A-7 shows the slider control properties editor:

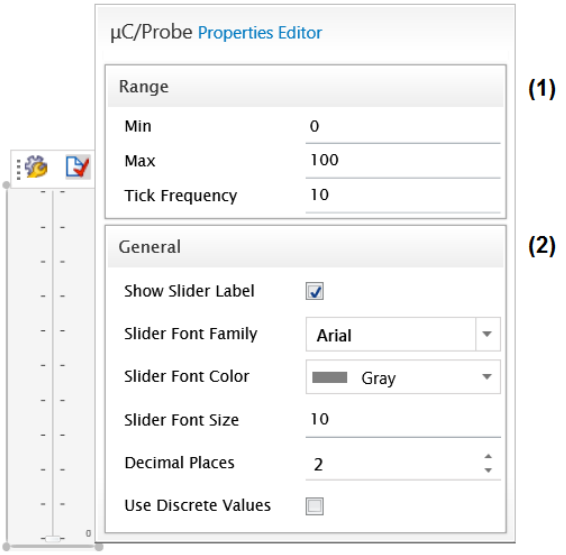

Figure A-7 **Slider Control Properties Editor**

FA-7(1) The slider control has a minimum, a maximum, and an increment value. The **Tick Frequency** not only determines the increment value but also the number of tick marks along the track.

FA-7(2) The **General** category includes the formatting properties that affect the slider's tick labels. Select the check box **Use Discrete Values** if you want the slider to adjust the associated symbol by making discrete increments.

A-2-2 CUSTOM SLIDER PROPERTIES EDITOR

The custom slider is similar to the one from section A-2-1, except that it also allows you to include two images to the left and right side of the slider's track as shown in Figure A-8:

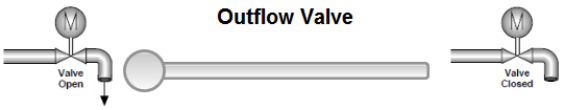

Figure A-8 **Custom Slider Example**

Imagine you want to control the state of a valve to be either open or closed by modifying an embedded target's application variable named `AppValveOutPct` that stores the state of the outflow valve (0%:open and 100%:closed). Figure A-9 shows the custom slider properties editor for such example:.

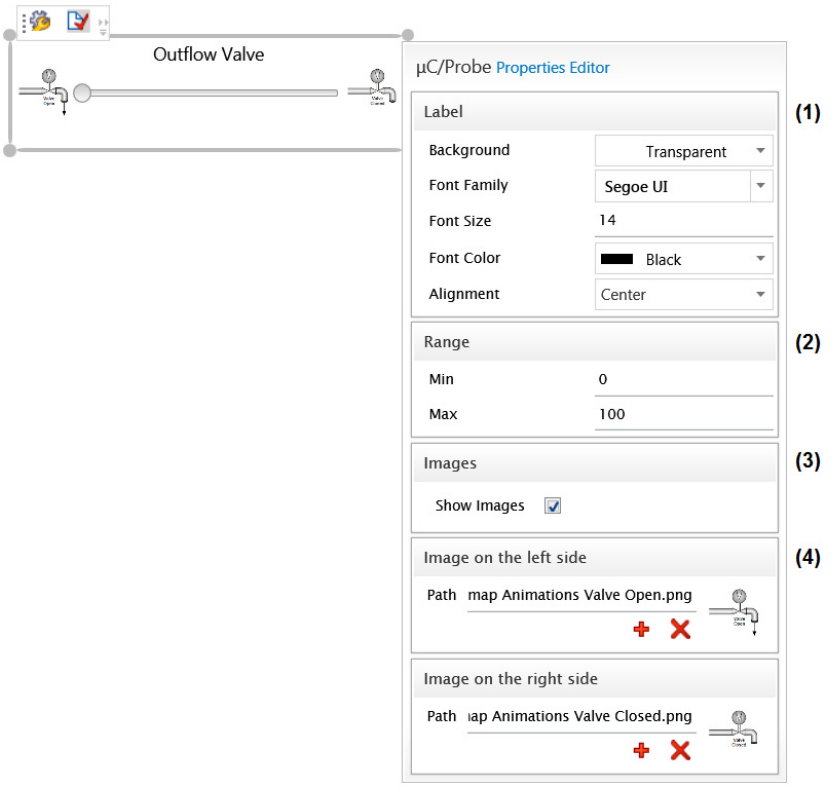

Figure A-9 **Custom Slider Properties Editor**

FA-9(1) The label in this example is Outflow Valve. The label properties include the background color, font styles and alignment.

FA-9(2) The custom slider control has a minimum and maximum range.

FA-9(3) You can turn on and off the images.

FA-9(4) Specify the path or browse to the bitmap you want to be placed on the left and right sides of the track.

A-2-3 CUSTOM SWITCH PROPERTIES EDITOR

The custom switch control is a two state button. You can modify the value of an embedded target's symbol by specifying the values you want to write when the button is switched between the **On** and **Off** states as shown in Figure A-10:

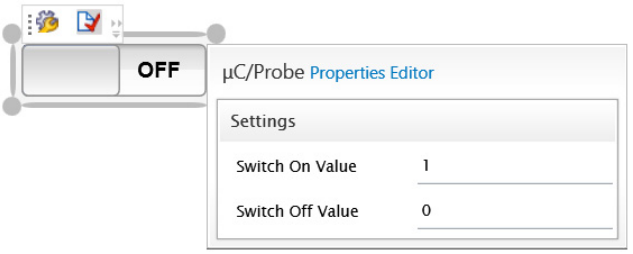

Figure A-10 **Custom Switch Properties Editor**

A-2-4 CHECKBOX PROPERTIES EDITOR

The checkbox control is similar to the custom switch but it also allows you to specify a label to display when the checkbox is selected and not selected. In the example shown in Figure A-11 such labels are set to Enabled during the On state and Disabled during the Off state.

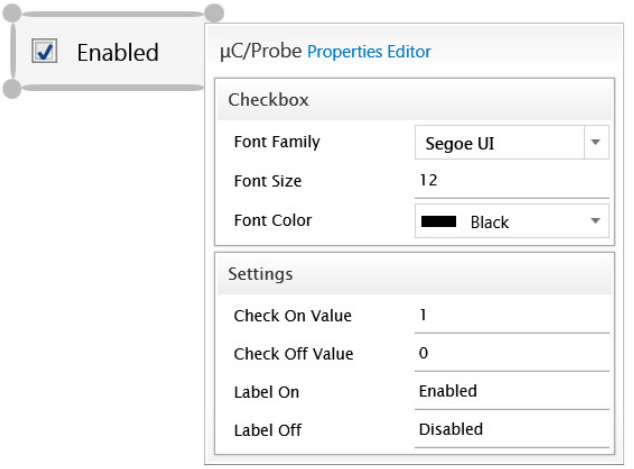

Figure A-11 **Checkbox Properties Editor**

A-2-5 PUSH BUTTON PROPERTIES EDITOR

The push button control is a momentary switch that switches between the states **On** while held down and **Off** when released. The properties window is shown in Figure A-12.

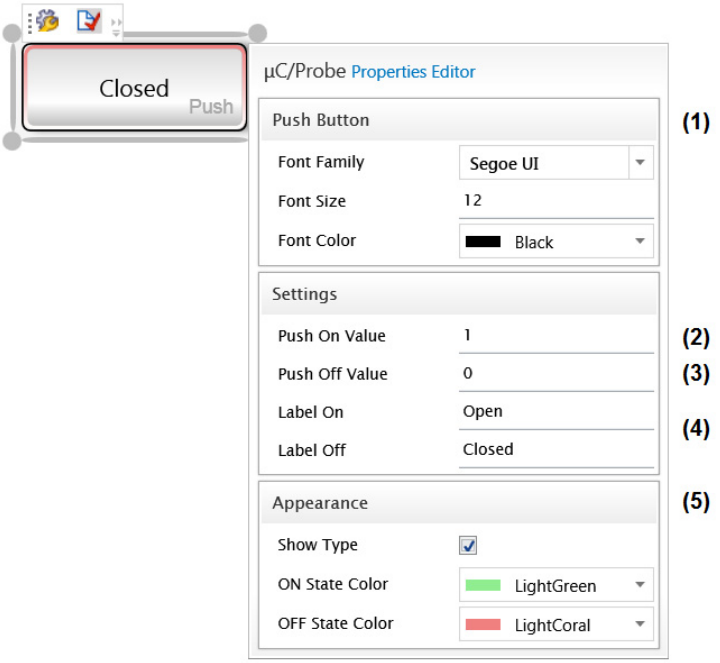

Figure A-12 **Push Button Properties Editor**

FA-12(1) You can specify the labels to display during the **On** and **Off** states. These properties affect the label's font family, font size and font color properties.

FA-12(2) During run-time, when the user holds down the push button, µC/Probe writes the **Push On Value** one single time to the embedded target.

FA-12(3) When the user releases the button, µC/Probe writes the **Push Off Value** one single time to the embedded target.

FA-12(4) Here you specify the labels you want to display during the **On** and **Off** states.

FA-12(5) You can also specify the border colors you want to display during the **On** and **Off** states and whether or not you want to show the **Push** label on the corner.

A-2-6 TOGGLE BUTTON PROPERTIES EDITOR

The toggle button control is a button that switches between the states **On** and **Off** when clicked. The properties window for a relay's toggle button is shown in Figure A-13.

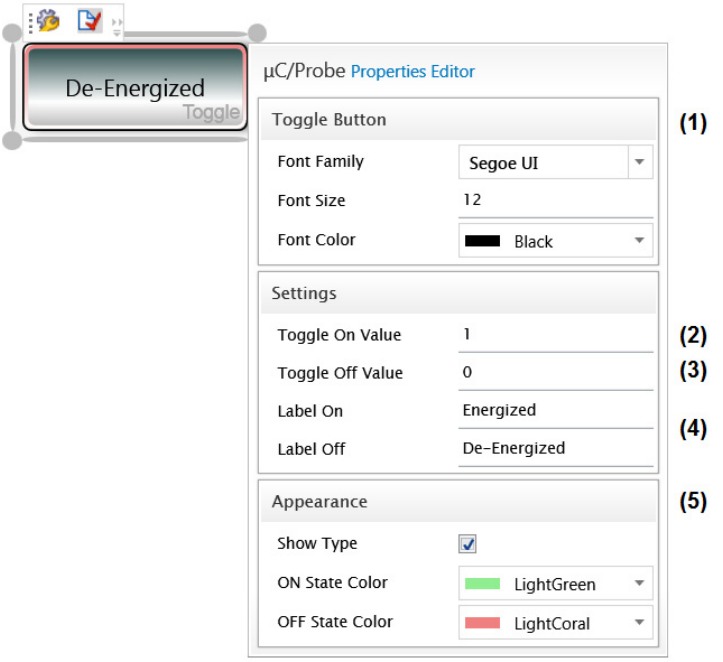

Figure A-13 **Toggle Button Properties Editor**

FA-13(1) You can specify the labels to display during the **On** and **Off** states. These properties affect the label's font family, font size and font color properties.

FA-13(2) During run-time, when the user clicks and releases the button, µC/Probe writes the **Toggle On Value** one single time to the embedded target.

FA-13(3) When the user clicks and releases the button again, µC/Probe writes the **Toggle Off Value** one single time to the embedded target.

FA-13(4) Here you specify the labels you want to display during the **On** and **Off** states.

FA-13(5) You can specify the border colors you want to display during the **On** and **Off** states and whether or not you want to show the **Toggle** label on the corner.

A-2-7 REPEAT BUTTON PROPERTIES EDITOR

The repeat button control is a button that switches between the states **On** while held down and **Off** when released. The properties window is shown in Figure A-14.

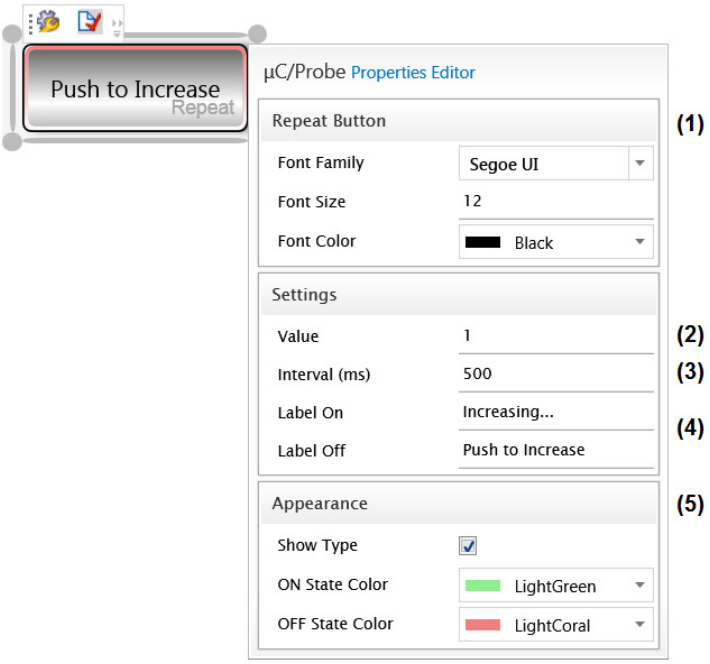

Figure A-14 **Repeat Button Properties Editor**

FA-14(1) You can specify the labels to display during the **On** and **Off** states. These properties affect the label's font family, font size and font color properties.

FA-14(2) During run-time, when the user holds down the repeat button, µC/Probe writes the **Value** multiple times to the embedded target until the button is released.

FA-14(3) The value gets written to the embedded target multiple times at the specified interval in milliseconds.

FA-14(4) Here you specify the labels you want to display during the **On** and **Off** states.

FA-14(5) You can specify the border colors you want to display during the **On** and **Off** states and whether or not you want to show the **Repeat** label on the corner.

A-3 CHARTS

Figure A-15 shows the three types of charts supported by µC/Probe:

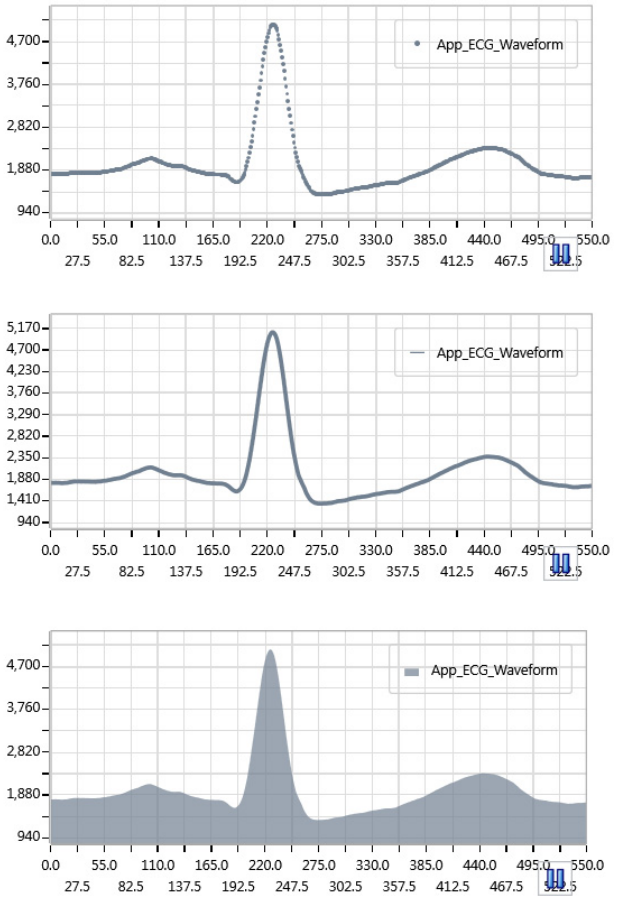

Figure A-15 **Marker, Line and Area Charts**

A-3-1 CHART PROPERTIES EDITOR

All three chart types share the same properties editor as shown in Figure A-16:

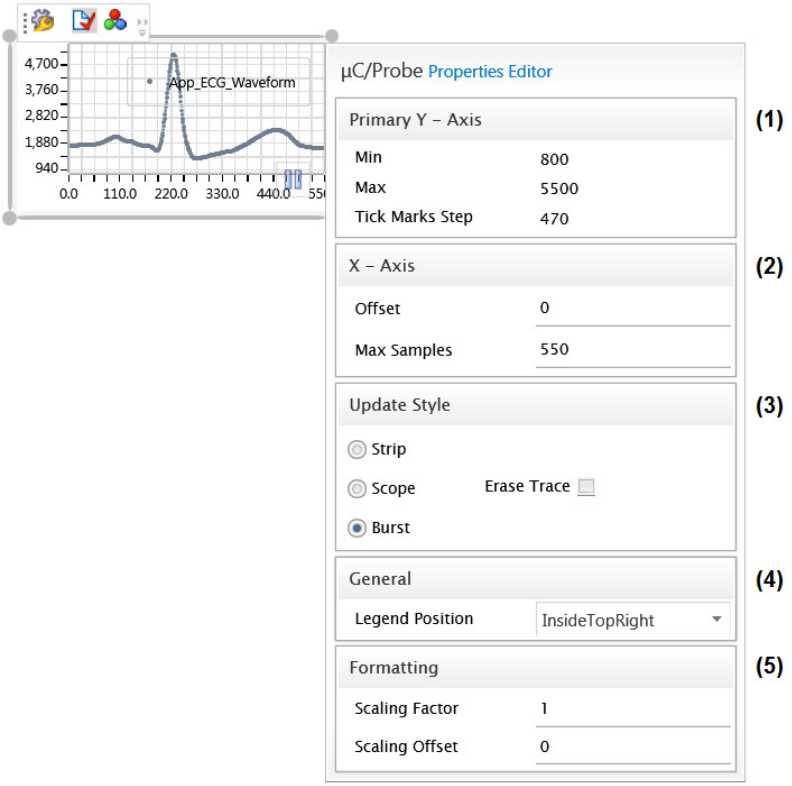

Figure A-16 **Charts Properties Editor**

FA-16(1) In order to change the number at which the vertical axis value starts or ends, enter a different number in the **Min** box or the **Max** box.

In order to change the interval of tick marks and chart grid lines, enter a different number in the **Tick Marks Step** box.

FA-16(2) In order to change the number at which the horizontal axis value starts or ends, enter a different number in the **Offset** box or the **Max Samples** box.

FA-16(3) µC/Probe updates charts in one of three modes:

- **Strip** Mode: similar to a chart recorder device that prints over a paper strip, µC/Probe first plots points from the left to the right side of the chart. From there, any new points are plotted at the rightmost side of the chart by shifting old points to the left.

- **Scope** Mode: similar to an oscilloscope, µC/Probe first plots points from the left to the right side of the chart. Once the plot reaches the right side of the chart, it erases the plot and begins plotting again from the left side of the chart.

- **Burst** Mode: this update mode was made for high performance applications where you want to plot array data quickly by plotting the entire array in one sweep. µC/Probe will not shift any points over the plotting area. Instead, it will erase the plot and will plot the same array again, assuming that the array is being updated by the embedded target.

FA-16(4) µC/Probe supports charts with multiple data series. That means that you can associate multiple symbols from your embedded target into one single chart.

In order to tell which trace represents a symbol in your chart, a color-coded legend with the name of the symbol is displayed over the chart. Select the legend position that better suits your needs.

FA-16(5) In case you need to convert the value points to Engineering Units (EU) before plotting in the chart, you can use the scaling factor and offset to specify the parameters of a linear conversion function. For example, if the embedded target's symbol you need to display is a 4-20mA value, you can implement the standard linear equation y = mx + b where m is the scaling factor, x is the 4-20mA value, b is the offset and y is the resulting Engineering Units (EU) value to be plotted.

A-3-2 CHART SERIES EDITOR

The charts series editor allows you to configure each trace in the plotting area. The series editor is shared by all three types of charts as shown in Figure A-17:

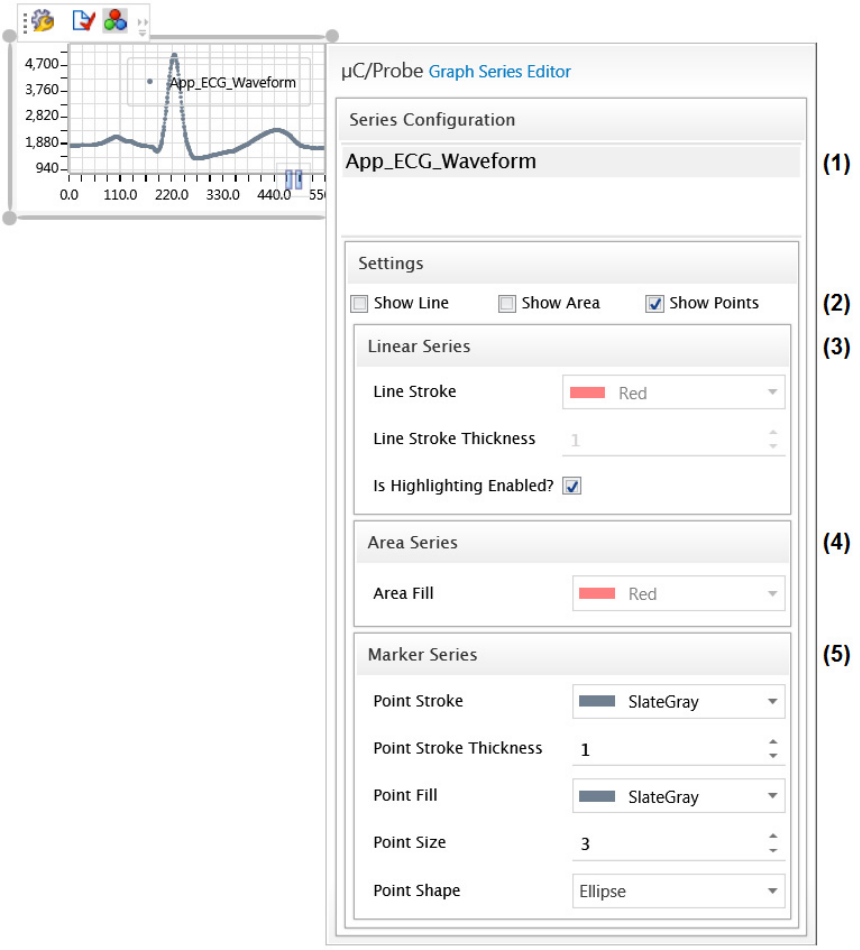

Figure A-17 **Charts Series Editor**

FA-17(1) Select the data series you want to work with. By default, the name of the data series is the same of the symbol as declared in your embedded target C files.

FA-17(2) µC/Probe supports three types of charts: marker, line and area charts. You can mix and match chart types in one single chart by selecting the corresponding check box.

FA-17(3) If the **Show Line** check box is selected, this section allows you to configure the color and thickness of the line.

FA-17(4) If the **Show Area** check box is selected, this section allows you to configure the color of the area.

FA-17(5) If the **Show Points** check box is selected, this section allows you to configure the color, thickness and shape of the points.

Appendix B

Kernel Awareness Screen

µC/Probe allows you to add pre-configured data screens that display some of the most popular Micriµm's software modules.

Figure B-1 and Figure B-2 show examples of the Kernel Awareness Screens for µC/OS-III. The Screens display µC/OS-III's internal data structures in a convenient series of windows. This provides you with information about each of the active tasks in the embedded target application among other kernel objects such as semaphores, event flags, mutexes, etc.

µC/OS-III KA | DataScreen1

CPU Usage 1.42 %

		µs
Miscellaneous		
CPU Usage (%)	1.42	
OS Running	Yes	
Idle Task Counter	22,740,203	
Statistic Task Counter	20,382	
Tick Task Counter	100,010	
Timer Task Counter	1,000	
#Context Switches	222,956	
Message Queue Pool		
#Free Messages	99	
#Used Messages	1	
#Used Messages (Max)	Feature Not Enabled	
µC/OS-III Task Execution Times		
Tick Task	1,144	15.9
Timer Task	108	1.5
Statistics Task	35,270	489.9
Interrupt Handler Task	Feature Not Enabled	
Interrupts		
Nesting Counter	0	
Maximum Interrupt Disable Time	1,186	16.5
Interrupt Queue		
#Entries	Feature Not Enabled	
#Entries (Max)	Feature Not Enabled	
#Overflows	Feature Not Enabled	
Scheduler		
Round Robin Enabled?	No	
Lock Nesting Counter	0	
Maximum Lock Time	508	7.1

Figure B-1 **Kernel Awareness Screen: Miscellaneous**

Figure B-2 shows the information displayed for each task. The columns can be sorted and the tri-color bar graphs highlight those tasks reaching their maximum stack space, which is a typical bug when developing embedded systems:

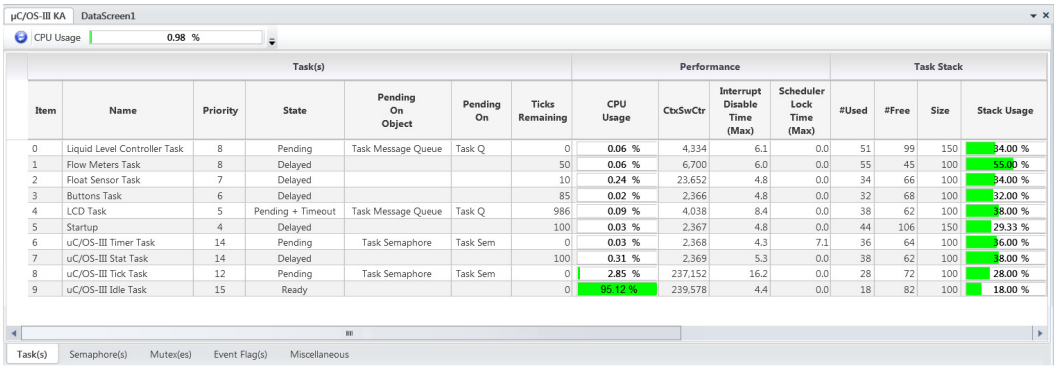
µC/OS-III KA
DataScreen1
CPU Usage
0.98 %
Task(s)
Performance
Task Stack
Item
Name
Priority
State
Pending On Object
Pending On
Ticks Remaining
CPU Usage
CtxSwCtr
Interrupt Disable Time (Max)
Scheduler Lock Time (Max)
#Used
#Free
Size
Stack Usage
0 Liquid Level Controller Task 8 Pending Task Message Queue Task Q 0 0.06 % 4,334 6.1 0.0 51 99 150 34.00 %
1 Flow Meters Task 8 Delayed 50 0.06 % 6,700 6.0 0.0 55 45 100 55.00 %
2 Float Sensor Task 7 Delayed 10 0.24 % 23,652 4.8 0.0 34 66 100 34.00 %
3 Buttons Task 6 Delayed 85 0.02 % 2,366 4.8 0.0 32 68 100 32.00 %
4 LCD Task 5 Pending + Timeout Task Message Queue Task Q 986 0.09 % 4,038 8.4 0.0 38 62 100 38.00 %
5 Startup 4 Delayed 100 0.03 % 2,367 4.8 0.0 44 106 150 29.33 %
6 uC/OS-III Timer Task 14 Pending Task Semaphore Task Sem 0 0.03 % 2,368 4.3 7.1 36 64 100 36.00 %
7 uC/OS-III Stat Task 14 Delayed 100 0.31 % 2,369 5.3 0.0 38 62 100 38.00 %
8 uC/OS-III Tick Task 12 Pending Task Semaphore Task Sem 0 2.85 % 237,152 16.2 0.0 28 72 100 28.00 %
9 uC/OS-III Idle Task 15 Ready 0 95.12 % 239,578 4.4 0.0 18 82 100 18.00 %
Task(s)
Semaphore(s)
Mutex(es)
Event Flag(s)
Miscellaneous

Figure B-2 **Kernel Awareness Screen: Task List**

Appendix

C

License and Activation

µC/Probe is distributed as shareware, you can download a trial version that allows you to create data screens with up to 10 symbols. You can use the µC/Probe trial version for an unlimited period of time to find out if it suits your needs.

The full version of µC/Probe allows an unlimited number of symbols and unlocks other more sophisticated features such as spreadsheets, that allow you to associate each cell with a symbol and use all the calculation features a regular spreadsheet has to offer.

C-1 ORDERING

If you decide to obtain the full version of µC/Probe, you will have to purchase a license.

The license key will unlock the trial version and entitle you to e-mail support and free minor software updates for an unlimited period of time.

Please contact us for further information about pricing and ordering options at:

Micriμm
1290 Weston Road, Suite 306
Weston, FL 33326

+1 954 217 2036
+1 954 217 2037 (FAX)

E-Mail : sales@micrium.com
Website : www.micrium.com

C-2 ACTIVATING

After you order a µC/Probe license, you will receive an e-mail containing your registration data and license code.

From your trial version of µC/Probe, click **File -> Register** and enter the license code in the activation dialog shown in Figure C-1:

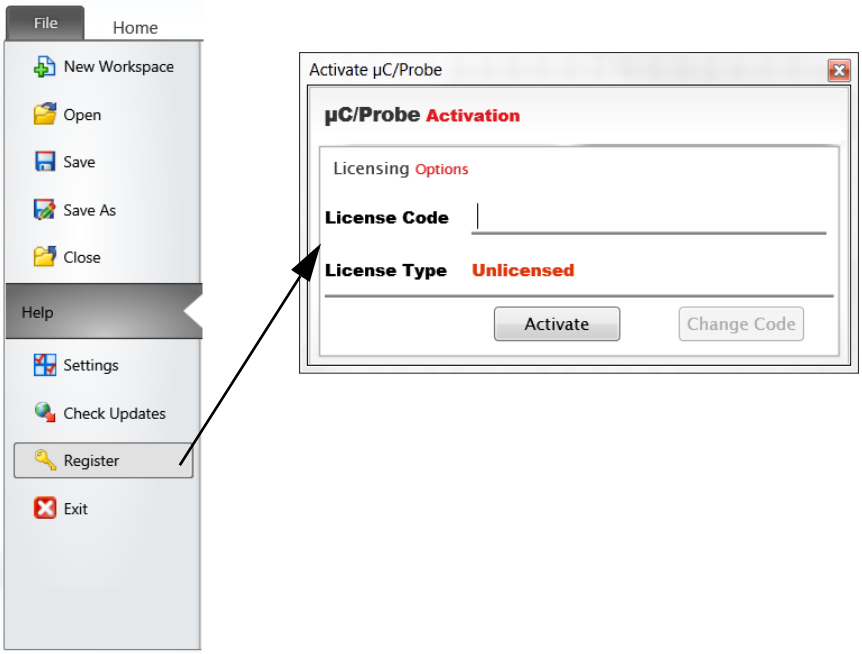

Figure C-1 **Activating the full version of µC/Probe**

Appendix

D

Bibliography

- Labrosse Jean. *µC/OS-II The Real-Time Kernel*. R&D Technical Books, ISBN 1-57820-103-9, 2002.

- Labrosse Jean. *µC/OS-III The Real-Time Kernel*. Micriµm Press, ISBN 978-0-98223375-3-0, 2009.

- Légaré Christian. *µC/TCP-IP The Embedded Protocol Stack*. Micriµm Press, 2011.

Index

A

activation 68
animation 48
animation properties editor 48–49
area charts 57

C

chart properties editor 58
chart series editor 60
charts 30, 57
charts properties editor 58
charts series editor 60
checkbox properties editor 53
circular gauges 28
communication settings 19
  J-Link 20
  RS-232 22
  TCP/IP 21
custom slider
  example 51
  properties editor 51
custom switch
  properties editor 52
cylinders 29

D

data flow 5
design time 10

E

ELF file 13
  browsing 14
example 35

F

formatting
  properties editor 46

G

general settings 18

H

half donuts 29
horizontal linear gauges 27

I

indicators 37

J

J-Link 20

K

kernel awareness screen
  miscellaneous 64
  task list 65

L

layout design tools 33
line charts 57
linear gauges 26

M

marker charts 57
miscellaneous (tools) 31

N

numeric indicator
  properties editor 48
numerics (tools) 30

O

ordering 67
output window 42
overview 7

P

properties editor
- animation 48–49
- checkbox 53
- custom slider 51
- formatting 46
- numeric indicator 48
- push button 54
- repeat button 56
- slider control 50
- toggle button 55

push button
- properties editor 54

Q

quadrant gauges 27

R

range and colors editor 47
repeat button
- properties editor 56

RS-232 22
run-time mode 42
run-time mode checklist 41

S

Segger J-Link 20
semicircle gauges 28
slider control
- properties editor 50

status bar 42
symbol browser
- loading an ELF file 13
- symbols grouped by C file 14

symbols 37
symbols manager 38

T

TCP/IP 21
toggle button
- properties editor 55

toolbar settings 17
toolbox 25
- charts 30
- circular gauges 28
- cylinders 29
- half donuts 29
- horizontal linear gauges 27
- linear gauges 26
- miscellaneous 31
- numerics 30
- quadrant gauges 27
- semicircle gauges 28
- writable controls 31

V

virtual controls 37, 45, 50
virtual indicators 45–46

W

Windows application 12
workspace explorer 23–24
writable controls 31

Z

µC/Probe data client 10
- design time 10